



**TUGAS AKHIR - TE141599**

**SISTEM MONITORING WAKTU PENDINGINAN  
RUANGAN UNTUK DETEKSI KEBERSIHAN FILTER AC**

Wisnu Baskoro  
NRP 2212100134

Dosen Pembimbing  
Suwito, ST., MT.  
Fajar Budiman, ST., MSc.

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**FINAL PROJECT - TE 141599**

**ROOM COOLING TIME MONITORING SYSTEM FOR  
AC FILTER CLEANLINESS DETECTION**

Wisnu Baskoro  
NRP 2212 100 134

Supervisor  
Suwito, ST., MT.  
Fajar Budiman, S.T., MSc.

ELECTRICAL ENGINEERING DEPARTMENT  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2017



## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “*Sistem Monitoring Waktu Pendinginan Ruangan untuk Deteksi Kebersihan Filter AC*” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2017

Wisnu Baskoro  
NRP 2212100134

*# Halaman ini sengaja dikosongkan #*

# **SISTEM MONITORING WAKTU PENDINGINAN RUANGAN UNTUK DETEKSI KEBERSIHAN FILTER AC**

## **TUGAS AKHIR**

**Diajukan untuk Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada**

**Bidang Studi Teknik Elektronika  
Jurusan Teknik Elektro  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya**

**Menyetujui:**

**Dosen Pembimbing I,**



**Suwito, ST., MT.**

**NIP. 198101052005011004**

**Dosen Pembimbing II,**



**Fajar Budiman, ST., M.Sc.**

**NIP. 198607072014041001**



**SURABAYA  
JANUARI, 2017**

*# Halaman ini sengaja dikosongkan #*



## **Sistem Monitoring Waktu Pendinginan Ruangan untuk Deteksi Kebersihan Filter AC**

**Nama** : Wisnu Baskoro  
**Pembimbing I** : Suwito, ST., MT.  
**Pembimbing II** : Fajar Budiman, ST., MSc.

### **ABSTRAK**

Tarif Dasar Listrik(TDL) semakin lama semakin naik. Konsumen harus lebih cermat untuk menghemat konsumsi listrik. Di sisi lain, kemampuan penyediaan energi listrik indonesia yang belum mencukupi kebutuhan masyarakat yang ditunjukkan dengan masih seringnya terjadi pemadaman bergilir juga memerlukan perhatian untuk kepentingan bersama. Pada umumnya konsumsi listrik paling besar dari sebuah bangunan adalah AC. Untuk itu diperlukan penggunaan AC yang cermat untuk merespon kedua masalah di atas. Salah satu cara menghemat Konsumsi listrik AC adalah dengan menjaga filter tetap bersih. Untuk itu dalam tugas akhir ini saya akan merancang pendeteksi kebersihan filter AC dengan mengukur daya AC menggunakan sensor ACS712 dan mengukur suhu ruangan menggunakan sensor LM35. Dari kedua variabel ini sistem dengan menggunakan mikrokontroler akan mampu menghitung waktu pendinginan ruangan dan mendeteksi kondisi kebersihan filter AC kemudian menampilkannya melalui HMI berupa simulasi pada komputer. kecepatan pendinginan AC untuk AC Haier HSU-05LEA03 saat kotor adalah  $841.8367\text{ s}^{\circ}\text{C}$ ,  $526.2295\text{ s}^{\circ}\text{C}$  saat bersih, dengan selisih  $315.6072\text{ s}^{\circ}\text{C}$ . Kecepatan pendinginan AC Sharp AH-AP5MHL dengan mode *fan lo* adalah  $1462,5\text{ s}^{\circ}\text{C}$  saat kotor dan  $621\text{ s}^{\circ}\text{C}$  saat bersih dengan selisih  $841.5\text{ s}^{\circ}\text{C}$ . Kecepatan pendinginan AC Sharp AH-AP5MHL dengan mode *fan med* adalah  $546\text{ s}^{\circ}\text{C}$  saat kotor dan  $358\text{ s}^{\circ}\text{C}$  saat bersih dengan selisih  $188\text{ s}^{\circ}\text{C}$ . Kecepatan pendinginan AC Sharp AH-AP5MHL dengan mode *fan hi* adalah  $246\text{ s}^{\circ}\text{C}$  saat kotor dan  $205.5\text{ s}^{\circ}\text{C}$  saat bersih dengan selisih  $40.5\text{ s}^{\circ}\text{C}$ .

**Kata Kunci:** ACS712, LM35, Mikrokontroler, Energi, Suhu, Filter AC

*# Halaman ini sengaja dikosongkan #*

## ***Room Cooling Time Monitoring System for AC Filter Cleanliness Detection***

**Name** : Wisnu Baskoro  
**Supervisor I** : Suwito, ST., MT.  
**Supervisor II** : Fajar Budiman, ST., MSc.

### **ABSTRACT**

Electricity fare is increasing by the time. Consumers should be more careful to save power consumption. On the other hand, Indonesia's ability of the electricity production is not sufficient given by people still experiencing frequent rolling blackouts also requires attention to the common interest. In general, most of the electricity consumption of a building is air conditioner. And there's usually more than one air conditioner in a building. It required careful usage of AC to respond to both of the problems mentioned above. One way to save on air conditioning electricity consumption is to keep the filter clean. Therefore, in this final project I will design the AC filter cleanliness detection by measuring the AC power using the ACS712 and measure the room temperature using LM35. From both of these variables by using microcontroller, the system will be able to calculate cooling time and detect the condition of cleanliness AC filter and display it via computer simulation. the ac cooling speed of AC Haier HSU-05LEA03 when dirty is  $841.8367 \text{ s}^{\circ}\text{C}$ ,  $526.2295 \text{ s}^{\circ}\text{C}$  when clean by margin of  $315.6072 \text{ s}^{\circ}\text{C}$ . AC cooling speed of Sharp AH-AP5MHL with lo fan mode is  $1462.5 \text{ s}^{\circ}\text{C}$  when dirty and  $621 \text{ s}^{\circ}\text{C}$  when clean by margin of  $841.5 \text{ s}^{\circ}\text{C}$ . AC cooling speed of Sharp AH-AP5MHL with med fan mode is  $546 \text{ s}^{\circ}\text{C}$  when dirty and  $358 \text{ s}^{\circ}\text{C}$  when clean by margin of  $188 \text{ s}^{\circ}\text{C}$ . AC cooling speed of Sharp AH-AP5MHL with hi fan mode is  $246 \text{ s}^{\circ}\text{C}$  when dirty and  $205.5 \text{ s}^{\circ}\text{C}$  when clean by margin of  $40.5 \text{ s}^{\circ}\text{C}$ .

**Keywords:** ACS712, LM35, Microcontroller, Energy, Temperature, AC Filter

*# Halaman ini sengaja dikosongkan #*

## KATA PENGANTAR

Puji syukur atas rahmat dan hidayah yang diberikan oleh Allah SWT. Karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan tugas akhir ini. Selama pelaksanaan penelitian Tugas Akhir ini, penulis mendapatkan bantuan dari berbagai pihak baik dukungan secara moril maupun materiil. Penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada berbagai pihak yang mendukung dan membantu dalam tugas akhir ini, diantaranya :

1. Allah SWT yang telah memlimpahkan ramhat dan hidayah-Nya serta memperlancar dalam pengerjaan Tugas Akhir ini.
2. Kedua orang tua tercinta, Bapak Doddy dan Ibu Rina yang tidak pernah putus untuk seluruh do'a, nasihat, motivasi, dan dukungannya.
3. Suwito, ST., MT. selaku dosen pembimbing pertama, atas bimbingan, inspirasi, pengarahan, dan motivasi yang diberikan selama pengerjaan penelitian tugas akhir ini.
4. Fajar Budiman, ST., MSc. selaku dosen pembimbing kedua, atas bimbingan, inspirasi, pengarahan, dan motivasi yang diberikan selama pengerjaan penelitian tugas akhir ini.
5. Teman-teman Elektro ITS maupun dari luar yang senantiasa membantu dan memberikan semangat dalam mengerjakan tugas akhir.
6. Semua pihak yang turut membantu dalam pengerjaan Tugas Akhir ini yang tidak dapat disebutkan satu per satu.

Penulis sadar bahwa Tugas Akhir ini masih belum sempurna dan masih banyak hal yang perlu diperbaiki. Saran, kritik dan masukan dari semua pihak sangat membantu penulis terutama untuk berbagai kemungkinan pengembangan tugas akhir ini.

Surabaya, Januari 2017

Penulis

*# Halaman ini sengaja dikosongkan #*

## DAFTAR ISI

<b>HALAMAN PENGESAHAN .....</b>	<b>iii</b>
<b>ABSTRAK .....</b>	<b>v</b>
<b>ABSTRACT .....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>ix</b>
<b>DAFTAR ISI.....</b>	<b>ix</b>
<b>LIST OF CONTENT .....</b>	<b>xi</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiii</b>
<b>DAFTAR TABEL .....</b>	<b>xv</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Permasalahan.....	2
1.3 Tujuan.....	2
1.4 Batasan Masalah.....	2
1.5 Metodologi .....	3
1.6 Sistematika Penulisan .....	4
1.7 Relevansi.....	4
<b>BAB II TINJAUAN PUSTAKA DAN DASAR TEORI .....</b>	<b>7</b>
2.1 Tinjauan Pustaka .....	7
2.1.1 Konservasi Energi.....	7
2.1.2 Teknologi Terkini .....	7
2.2 Dasar Teori.....	8
2.2.1 Sensor Arus.....	9
2.2.2 <i>Hall Effect</i> .....	10
2.2.3 Sensor Suhu.....	11
2.2.4 <i>Thermistor</i> .....	11
2.2.5 Mikrokontroler .....	12
2.2.6 ADC.....	13
2.2.7 Ethernet.....	14
2.2.8 Protokol ModBus.....	15
2.2.9 HMI .....	17
2.2.10 Kalor .....	17
2.2.11 AC .....	18
<b>BAB III PERANCANGAN SISTEM.....</b>	<b>21</b>
3.1 Diagram Blok Sistem.....	21

3.2	Perancangan Perangkat Keras ( <i>Hardware</i> ).....	22
3.2.1	Perancangan Sensor Arus .....	23
3.2.2	Perancangan Sensor Suhu.....	25
3.2.3	Rangkaian Mikrokontroler.....	26
3.2.4	Rangkaian HMI .....	29
3.2.5	Perangkat yang Dimonitor .....	29
3.3	Perancangan Perangkat Lunak ( <i>Software</i> ) .....	31
3.3.1	Perancangan Program Mikrokontroler .....	31
3.3.2	Pemrograman HMI .....	35
<b>BAB IV</b>	<b>PENGUJIAN DAN ANALISIS .....</b>	<b>37</b>
4.1	Pengukuran Arus.....	39
4.2	Pengukuran Suhu .....	42
4.3	Pengujian Sistem.....	43
4.3.1	Pengukuran Ketika AC Kotor.....	44
4.3.2	Pengukuran Ketika AC Bersih.....	48
4.4	Pengujian HMI.....	52
4.5	Pengujian Menggunakan AC <i>Portable</i> .....	54
<b>BAB V</b>	<b>PENUTUP .....</b>	<b>57</b>
5.1	Kesimpulan.....	57
5.2	Saran .....	58
<b>DAFTAR PUSTAKA</b>	<b>.....</b>	<b>59</b>
<b>LAMPIRAN</b>	<b>.....</b>	<b>61</b>
<b>BIODATA PENULIS</b>	<b>.....</b>	<b>65</b>



## LIST OF CONTENT

<b>APPROVAL SHEET .....</b>	<b>iii</b>
<b>ABSTRAK .....</b>	<b>v</b>
<b>ABSTRACT .....</b>	<b>vii</b>
<b>PREFACE .....</b>	<b>ix</b>
<b>DAFTAR ISI .....</b>	<b>ix</b>
<b>LIST OF CONTENT .....</b>	<b>xi</b>
<b>LIST OF IMAGE .....</b>	<b>xiii</b>
<b>LIST OF TABLE .....</b>	<b>xv</b>
<b>BAB I INTRODUCTION .....</b>	<b>1</b>
1.1 Background .....	1
1.2 Problem .....	2
1.3 Aim of Project .....	2
1.4 Scope of Problem .....	2
1.5 Methodology .....	3
1.6 Writing Systematic .....	4
1.7 Relevance .....	4
<b>BAB II LITERATURE AND THEORY .....</b>	<b>7</b>
2.1 Literature .....	7
2.1.1 Energy Conservation .....	7
2.1.2 Recent Technology .....	7
2.2 Theory .....	8
2.2.1 Current Sensor .....	9
2.2.2 Hall Effect .....	10
2.2.3 Temperature Sensor .....	11
2.2.4 Thermistor .....	11
2.2.5 Microcontroller .....	12
2.2.6 ADC .....	13
2.2.7 Ethernet .....	14
2.2.8 ModBus Protocol .....	15
2.2.9 HMI .....	17
2.2.10 Heat .....	17
2.2.11 AC .....	18
<b>BAB III SYSTEM DESIGN .....</b>	<b>21</b>
3.1 System Block Diagram .....	21

3.2	Hardware Design .....	22
3.2.1	Current Sensor Design .....	23
3.2.2	Temperature Sensor Design .....	25
3.2.3	Microcontroller Schematic .....	26
3.2.4	HMI Design .....	29
3.2.5	Monitored Device .....	29
3.3	Software Design.....	31
3.3.1	Programming Microcontroller .....	31
3.3.2	Programming HMI.....	35
<b>BAB IV</b>	<b>TESTING AND ANALYSIS .....</b>	<b>37</b>
4.1	Current Measurement.....	39
4.2	Temperature Measurement .....	42
4.3	System Testing.....	43
4.3.1	Measurement of Dirty AC.....	44
4.3.2	Measurement of clean AC .....	48
4.4	HMI Testing .....	52
4.5	System Testing Using Portable AC.....	54
<b>BAB V</b>	<b>CLOSURE .....</b>	<b>57</b>
5.1	Conclusion.....	57
5.2	Suggestion .....	58
<b>BIBLIOGRAPHY</b>	<b>.....</b>	<b>59</b>
<b>APPENDIX</b>	<b>.....</b>	<b>61</b>
<b>BIOGRAPHY</b>	<b>.....</b>	<b>65</b>

## DAFTAR GAMBAR

<b>Gambar 2.1</b> CleanAlert.....	8
<b>Gambar 2.2</b> Ilustrasi <i>Hall Effect</i> .....	10
<b>Gambar 2.3</b> Ilustrasi Perubahan Resistansi PTC dan NTC .....	12
<b>Gambar 2.4</b> Arsitektur Mikrokontroler.....	13
<b>Gambar 2.5</b> Ilustrasi I/O ADC 3bit .....	14
<b>Gambar 2.6</b> Hubungan <i>Request-Response</i> pada divais ModBus ....	17
<b>Gambar 3.1</b> Diagram Blok Sistem.....	22
<b>Gambar 3.2</b> Diagram Blok <i>Hardware</i> .....	23
<b>Gambar 3.3</b> Rangkaian Sensor Arus.....	24
<b>Gambar 3.4</b> Rangkaian Umum ACS712.....	24
<b>Gambar 3.5</b> Karakteristik Input Output ACS712 .....	25
<b>Gambar 3.6</b> <i>Basic Centigrade Temperature Sensor</i> .....	26
<b>Gambar 3.7</b> Rangkaian Sensor Suhu .....	26
<b>Gambar 3.8</b> Rangkaian Mikrokontroler .....	28
<b>Gambar 3.9</b> Diagram Blok HMI.....	29
<b>Gambar 3.10</b> Haier HSU-05LEA03 .....	30
<b>Gambar 3.11</b> Sharp AH-AP5MHL.....	30
<b>Gambar 3.12</b> Midea MPG-09CR.....	31
<b>Gambar 3.13</b> <i>Flow Chart</i> Program Arduino .....	34
<b>Gambar 3.14</b> Pemrograman <i>Interface</i> .....	35
<b>Gambar 3.15</b> Pemrograman <i>Interface</i> (2) .....	36
<b>Gambar 3.16</b> Pemrograman <i>Interface</i> (3) .....	36
<b>Gambar 4.1</b> Rangkaian Sistem .....	37
<b>Gambar 4.2</b> HMI Sistem.....	37
<b>Gambar 4.3</b> Haier HSU-05LEA03 .....	38
<b>Gambar 4.4</b> Sharp AH-AP5MHL.....	38
<b>Gambar 4.5</b> Midea MPG-09CR .....	38
<b>Gambar 4.6</b> Ruangn Buatan .....	39
<b>Gambar 4.7</b> Grafik Pengukuran Arus .....	41
<b>Gambar 4.8</b> Grafik penghitungan Arus oleh Sistem.....	41
<b>Gambar 4.9</b> Pengukuran Suhu oleh Sistem.....	42
<b>Gambar 4.10</b> Pengukuran Suhu oleh Termometer Ruangn .....	42
<b>Gambar 4.11</b> <i>Tapping</i> ACS712 pada AC.....	43
<b>Gambar 4.12</b> Pengukuran Arus dan Suhu oleh Sistem .....	44
<b>Gambar 4.13</b> Data Pengukuran Melalui <i>Serial Print</i> .....	44
<b>Gambar 4.14</b> Pengukuran Sistem Saat AC Kotor.....	45
<b>Gambar 4.15</b> Pengukuran Sistem Saat AC Kotor(2) .....	46

<b>Gambar 4.16</b> Pengukuran Sistem Saat AC Kotor(3).....	46
<b>Gambar 4.17</b> Pengukuran Sistem Saat AC Kotor(4).....	47
<b>Gambar 4.18</b> Filter AC Haier Kotor .....	47
<b>Gambar 4.19</b> Filter AC Sharp Kotor .....	48
<b>Gambar 4.20</b> Pengukuran Sistem Saat AC Bersih .....	49
<b>Gambar 4.21</b> Pengukuran Sistem Saat AC Bersih(2).....	50
<b>Gambar 4.22</b> Pengukuran Sistem Saat AC Bersih(3).....	50
<b>Gambar 4.23</b> Pengukuran Sistem Saat AC Bersih(4).....	51
<b>Gambar 4.24</b> Proses Pencucian AC .....	51
<b>Gambar 4.25</b> Filter AC Bersih.....	52
<b>Gambar 4.26</b> <i>Online Simulation</i> .....	52
<b>Gambar 4.27</b> <i>Online Simulation</i> (2).....	53
<b>Gambar 4.28</b> <i>Online Simulation</i> (3).....	53
<b>Gambar 4.29</b> Filter Bersih.....	54
<b>Gambar 4.30</b> Filter Diberi Kapas.....	55
<b>Gambar 4.31</b> Grafik Pengukuran Filter AC Bersih.....	55
<b>Gambar 4.32</b> Grafik Pengukuran Filter AC diberikan Kapas .....	56

## DAFTAR TABEL

<b>Tabel 4.1</b> Pengukuran Arus.....	40
<b>Tabel 4.2</b> Pengukuran Suhu.....	43
<b>Tabel 4.3</b> Pembulatan Suhu.....	43

*# Halaman ini sengaja dikosongkan #*

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Konsumsi energi final di Indonesia meningkat dari 778 juta SBM(Setara Barel Minyak) pada tahun 2000 menjadi 1.211 juta SBM pada tahun 2013 atau tumbuh rata-rata sebesar 3,46% per tahun. Pada tahun 2013 pangsa terbesar penggunaan energi adalah sektor industri (37,17%) diikuti oleh sektor rumah tangga (29,43%), transportasi (28,10%), komersial (3,24%), dan lainnya (2,04%)[1]. Rumah tangga menempati posisi kedua dengan 29,43% dari jumlah konsumsi listrik di Indonesia. Di sisi lain jumlah penduduk di Indonesia juga terus bertambah. Hal ini tentu akan berdampak pada jumlah permintaan listrik sektor rumah tangga khususnya rumah tangga dalam bentuk bangunan apartemen.

Di tengah terus meningkatnya kebutuhan listrik rumah tangga, selain diperlukannya tambahan jumlah energi listrik, diperlukan juga perubahan pola konsumsi listrik rumah tangga agar kebutuhan dan energi yang disediakan mampu seimbang. Salah satu penyebab tingginya konsumsi listrik pada sektor rumah tangga adalah tingginya daya yang diperlukan oleh pendingin ruangan. Jika energi listrik dari sektor rumah tangga khususnya pendingin ruangan tidak terbuang sia-sia maka energi listrik ini tentunya dapat lebih tepat guna.

Dalam rumah tangga pendingin ruangan merupakan salah satu alat yang menggunakan banyak daya listrik selain kulkas, oven, atau alat lainnya yang melibatkan elemen pemanas atau motor listrik. Jika alat lainnya pada umumnya hanya dibutuhkan satu buah untuk setiap kamar apartemen, lain halnya dengan pendingin ruangan yang pada setiap kamarnya bisa terdapat lebih dari satu buah. Oleh karena itu, biasanya pendingin ruangan merupakan alat listrik rumah tangga yang paling banyak memakan daya listrik. Pendingin ruangan memiliki sebuah filter udara yang berfungsi untuk memfilter kotoran udara.

Pemeliharaan yang paling penting yang akan menjamin efisiensi AC adalah dengan rutin mengganti atau membersihkan filter-nya. Karena apabila filter tersumbat, filter kotor akan menghambat aliran udara normal dan mengurangi efisiensi sistem secara signifikan. Dengan aliran udara normal terhambat, udara yang melewati filter dapat

membawa kotoran langsung ke kumparan *evaporator* dan menurunkan kemampuan menyerap panas dari kumparan[2]

Sebuah studi kasus di amerika yang dilakukan pada sebuah bangunan 34 lantai menunjukkan perawatan pembersihan pendingin ruangan sentral dapat meningkatkan efisiensi dari kumparan pendingin hingga 25%. Dengan bangunan 34 lantai perawatan yang tepat bahkan bisa menghemat pengeluaran bangunan hingga mencapai \$40,000 dalam setahun[3].

Dari permasalahan yang disampaikan di atas, rancang bangun sistem monitoring daya pendingin ruangan ini diharapkan mampu membantu pemilik rumah untuk mendeteksi kapan diperlukan pembersihan filter pendingin ruangan agar pemilik rumah dapat mengontrol konsumsi listrik dari rumahnya tanpa harus membuka unit pendingin ruangan.

## **1.2 Perumusan Masalah**

Berdasarkan latar belakang, dapat dirumuskan beberapa permasalahan. Permasalahan tersebut adalah sebagai berikut:

1. Membedakan jumlah waktu yang digunakan oleh AC untuk mendinginkan ruangan saat filter bersih dan kotor.
2. Mendeteksi ketika filter AC kotor
3. Membuat peringatan jika filter kotor

## **1.3 Tujuan Tugas Akhir**

Penelitian pada tugas akhir ini memiliki tujuan sebagai berikut:

1. Mendapatkan perbedaan Jumlah waktu yang digunakan oleh AC saat untuk mendinginkan ruangan filter bersih dan kotor.
2. Merealisasikan sistem deteksi kebersihan filter AC sebagai alat bantu penghemat energi.

## **1.4 Batasan Masalah**

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Sensor yang digunakan untuk mengukur daya adalah ACS712 untuk sensor arus.
2. Sensor yang digunakan untuk mengukur suhu ruangan adalah LM35.
3. Mikrokontroler yang digunakan adalah Arduino Uno.
4. HMI yang digunakan berupa simulasi *touchscreen*.
5. AC yang digunakan adalah AC split non *inverter*



## **1.5 Metodologi Penelitian**

Metodologi penelitian berisi mengenai tahapan-tahapan dalam pelaksanaan tugas akhir. Metodologi dari tugas akhir ini terdiri dari studi literatur, perancangan *hardware*, perancangan *software*, pengujian sistem, dan penulisan laporan tugas akhir.

### **1.5.1 Studi Literatur**

Pada tahap studi literatur dilakukan pengumpulan dasar-dasar teori yang mendukung proses penulisan tugas akhir. Pengumpulan dilakukan dari artikel, jurnal, paten, dan forum ilmiah. Studi literatur juga meliputi:

- Mempelajari pengukuran daya dan memilih sensor.
- Mempelajari pemrograman mikrokontroler dan HMI.

### **1.5.2 Perancangan *Hardware***

Setelah mendapatkan landasan teori yang cukup, langkah selanjutnya adalah merancang sistem yang nantinya akan diimplementasikan pada hardware. Pada tahap ini juga dilakukan survey komponen-komponen yang sesuai untuk sistem yang akan dibuat. Perancangan *hardware* ini meliputi perancangan sensor tegangan, perancangan sensor arus, perancangan sensor suhu, dan menghubungkan mikrokontroler dengan sensor-sensor dan HMI.

### **1.5.3 Perancangan *Software***

Pada tahap perancangan *Software* ini akan dibuat program yang membuat semua *Hardware* dapat berfungsi dan berkomunikasi. Poin pertama *software* berisi program yang berfungsi agar mikrokontroler dapat membaca data dari sensor arus, dan sensor suhu. Poin kedua *software* juga berisi program yang berfungsi agar HMI dapat menampilkan data dan juga dapat memberikan perintah dan input data kepada mikrokontroler. Poin ketiga *Software* berfungsi agar mikrokontroler dan HMI dapat saling berkomunikasi.

### **1.5.4 Pengujian Sistem**

Pada tahap ini sistem akan dijalankan dan dites keandalan dari, sensor arus, sensor suhu, program mikrokontroler, program HMI, dan komunikasi mikrokontroler dengan HMI. Jika belum didapatkan hasil yang sesuai maka akan dilakukan *troubleshoot*.

### **1.5.5 Penulisan Laporan Tugas Akhir**

Penulisan Laporan tugas Akhir beriringan dengan berjalannya kegiatan Tugas Akhir.

## **1.6 Sistematika Penulisan**

Sistematika penulisan berisi bab-bab yang ada di dalam buku. Bab-bab tersebut adalah pendahuluan, tinjauan pustaka dan dasar teori, perancangan sistem, pengujian dan analisis, dan penutup.

### **1.6.1 Bab 1: Pendahuluan**

Bab ini meliputi latar belakang, perumusan masalah, tujuan tugas akhir, metodologi penelitian, sistematika penulisan, dan relevansi.

### **1.6.2 Bab 2: Tinjauan Pustaka Dasar Teori**

Bab ini meliputi teori-teori penunjang dalam pelaksanaan tugas akhir ini, diantaranya teori tentang daya, teori tentang sensor tegangan, teori tentang sensor arus, teori tentang sensor suhu, teori tentang mikrokontroler, dan teori tentang HMI.

### **1.6.3 Bab 3: Perancangan Sistem**

Bab ini memaparkan langkah-langkah dalam merancang dan membuat sistem secara menyeluruh, dimulai dari perancangan *hardware*, membuat *hardware*, dan membuat *Software* untuk mikrokontroler dan HMI.

### **1.6.4 Bab 4: Pengujian dan Analisis**

Bab ini memaparkan hasil dari pengujian terhadap sistem dan analisa terhadap data hasil pengujian sistem.

### **1.6.5 Bab 5: Penutup**

Bab ini berisi kesimpulan dari seluruh rangkaian kegiatan tugas akhir, dan juga berisi saran-saran berupa hal-hal yang masih dapat dikembangkan lebih lanjut atau hal-hal yang belum sempat dikerjakan.

## **1.7 Relevansi atau Manfaat**

Hasil dari tugas akhir ini diharapkan dapat membantu penghematan energi rumah tangga dengan memonitor daya AC yang digunakan dan mendeteksi kebersihan filter AC. Selanjutnya diharapkan

hasil dari tugas akhir ini dapat memberikan inspirasi untuk penelitian-  
penelitian berikutnya.

*# Halaman ini sengaja dikosongkan #*

## **BAB 2**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Tinjauan pustaka memaparkan hal-hal yang berkaitan dengan tugas akhir ini dan juga teknologi yang sudah ada.

##### **2.1.1 Konservasi Energi**

Menurut kamus bahasa Indonesia, konservasi berarti pemeliharaan dan perlindungan sesuatu secara teratur untuk mencegah kerusakan dan kemusnahan dengan jalan mengawetkan/pengawetan. Berdasarkan SNI 03-6389-2000, konservasi energi adalah upaya mengefisienkan pemakaian energi untuk suatu kebutuhan agar pemborosan energi dapat dihindarkan. Konservasi energi disini mempunyai arti melestarikan/menghemat penggunaan energi listrik yang berasal dari sumber energi yang tidak dapat diperbaharui antara lain energi fosil-minyak bumi, batubara dan gas bumi. Konservasi energi merupakan salah satu langkah yang dapat dilakukan untuk dapat menghemat penggunaan sumber daya fosil yang semakin menipis[4].

Konservasi energi juga memiliki arti penggunaan energi secara efisien tanpa mengurangi penggunaan energi yang memang dibutuhkan. Sehingga penggunaan energi menjadi lebih hemat tetapi kebutuhan akan energi tetap terpenuhi. Dapat disimpulkan konservasi energi adalah upaya untuk menghemat energi dengan cara mencegah terbuangnya energi dari penggunaan yang tidak dibutuhkan.

Konservasi energi dapat dilakukan pada setiap lini dari rantai energi. Konservasi energi pada sisi sumber daya energi dapat dilakukan dengan mengurangi penggunaan sumber daya energi yang tidak terbarukan dan beralih ke sumber daya energi yang terbarukan. Konservasi energi pada sisi teknologi dapat dilakukan dengan penggunaan teknologi yang efisiensinya tinggi. Teknologi yang dimaksud mencakup teknologi dari proses eksploitasi sumber daya energi, konversi energi, distribusi energi, sampai teknologi yang menggunakan energi akhir.

##### **2.1.2 Teknologi Terkini**

Saat ini sebagian besar masyarakat khususnya pengguna AC masih belum menyadari pentingnya membersihkan filter AC di waktu

yang tepat. Sebagian besar pengguna AC hanya membersihkan filter dan AC berdasarkan anjuran perawatan AC yang diberikan oleh produsen AC. Faktanya di lapangan setiap ruangan memiliki tingkat kebersihan, kelembapan, dan faktor-faktor lainnya yang menyebabkan filter AC kotor yang berbeda.

Sementara itu di negara maju seperti amerika masyarakatnya sudah memiliki kesadaran yang lebih baik akan pentingnya menjaga kebersihan filter AC. Kesadaran ini timbul karena adanya penelitian yang dilakukan oleh pemerintahnya dan karena sebagian besar penduduk yang tinggal di kota besar tinggal pada sebuah bangunan bersama seperti apartemen yang membuat konsumsi energi listrik dalam suatu wilayah sangat terasa.

Saat ini di Amerika sudah beredar alat pendeteksi kebersihan filter AC yang bernama CleanAlert. CleanAlert ini bekerja dengan cara mendeteksi perbedaan tekanan udara sebelum udara melalui filter AC dan setelah udara melalui filter AC.



**Gambar 2.1** CleanAlert[5]

## **2.2 Dasar Teori**

Dasar teori berisi teori-teori penunjang mengenai sistem yang akan dibangun.

### 2.2.1 Sensor Arus

Sensor arus adalah sebuah alat yang mendeteksi besaran berupa arus listrik dan merubah besaran tersebut menjadi sinyal keluaran yang proporsional terhadap besaran arus yang dideteksi. Sinyal keluaran dari sensor arus dapat berupa tegangan, arus, ataupun digital. sensor arus dapat digunakan untuk amperemeter sebagai pengukur atau sebagai bagian dari sebuah sistem yang memiliki fungsi tertentu.

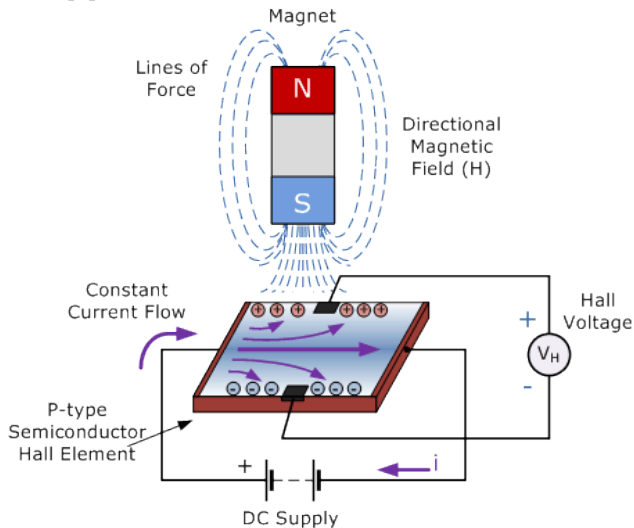
Sensor arus dapat mengukur arus searah maupun arus bolak-balik. Karakteristik sinyal keluaran dari sensor arus berbeda-beda sesuai dengan jenis arus yang diukur. Untuk mengukur arus searah sensor arus memiliki dua jenis sinyal keluaran, yaitu analog dan digital. Sinyal keluaran analog menghasilkan sinyal yang menduplikasi bentuk sinyal arus yang diukur. Sedangkan sinyal keluaran digital menghasilkan sinyal berupa data digital sesuai nilai arus yang diukur. Untuk mengukur arus bolak-balik sensor arus memiliki dua jenis sinyal keluaran, yaitu bipolar dan unipolar. Sinyal keluaran bipolar menghasilkan sinyal yang menduplikasi bentuk sinyal arus yang diukur. Sedangkan sinyal keluaran unipolar menghasilkan sinyal berupa nilai RMS dari arus bolak-balik yang diukur.

Sensor arus dibagi menjadi beberapa jenis berdasarkan cara kerjanya. Jenis sensor arus yang pertama adalah sensor *hall effect*. Sensor *hall effect* bekerja dengan cara mendeteksi medan magnet yang dihasilkan oleh arus. Jenis sensor arus yang kedua adalah sensor *current clamp*. Sensor *current clamp* bekerja dengan menggunakan prinsip transformator. Sensor *current clamp* hanya bekerja untuk pengukuran arus bolak-balik. Jenis sensor arus yang ketiga adalah sensor resistif. Sensor resistif bekerja dengan cara memberikan sebuah tahanan kepada arus yang akan menghasilkan tegangan yang proporsional terhadap arus. Jenis sensor arus yang keempat adalah sensor arus fiber optik. Sensor arus fiberoptik bekerja dengan menggunakan interferometer untuk mendeteksi perubahan fasa pada cahaya yang diproduksi oleh medan magnet. Sensor arus fiber optik hanya bisa digunakan untuk mengukur arus searah dan biasanya diaplikasikan untuk mengukur arus yang tinggi. Jenis sensor arus yang kelima adalah sensor Rogowski Coil. Rogowski Coil digunakan untuk mengukur arus bolak-balik dan memiliki kelebihan dapat mengukur pulsa arus dengan kecepatan tinggi.

### 2.2.2 Hall Effect

Sensor *hall effect* adalah sebuah alat yang mendeteksi medan magnet dan menghasilkan sinyal keluaran berupa tegangan yang proporsional dengan medan magnet. Pada umumnya sensor yang bekerja menggunakan medan magnet memiliki keterbatasan berupa tidak bisa digunakan apabila medan magnet statis. Berbeda untuk sensor *hall effect*, sensor *hall effect* dapat mengukur medan magnet yang statis.

Sensor *hall effect* terdiri dari sebuah lapisan silikon yang berfungsi untuk mengalirkan arus listrik. Pada ujung masing-masing dari lapisan silikon yang tegak lurus terhadap arah arus listrik diletakkan elektroda. Apabila tidak terdapat medan magnet yang mempengaruhi lapisan silikon maka ujung-ujung lapisan silikon yang diletakkan elektroda tidak akan memiliki perbedaan potensial tegangan. Ketika terdapat medan magnet yang mempengaruhi lapisan silikon maka arus yang mengalir pada lapisan silikon akan berbelok mendekati/menjauhi sisi yang lapisan silikon yang diberi medan magnet. Berbeloknya arus di dalam lapisan silikon akan menyebabkan perbedaan potensial tegangan pada ujung-ujung silikon yang diletakkan elektroda. Besarnya medan magnet akan menghasilkan perbedaan potensial tegangan yang proporsional[6].



**Gambar 2.2** Ilustasi *Hall Effect*[7]



### 2.2.3 Sensor Suhu

Sensor suhu adalah sebuah alat yang berfungsi membaca besaran suhu dan mengubahnya menjadi besaran lain untuk mencapai tujuan tertentu. Tujuan disini dapat berupa sebagai saklar pemutus atau untuk kepentingan pembacaan nilai. Sensor suhu yang berfungsi untuk membaca nilai memiliki dua jenis sinyal keluaran, yaitu sinyal analog dan sinyal digital.

Sensor suhu dapat beroperasi dengan dua metode, yaitu kontak fisik maupun non-kontak. Sensor suhu kontak fisik memerlukan sensor tersebut untuk melakukan kontak langsung dengan objek yang akan diukur suhunya. Objek yang diukur dapat berupa zat padat, cair, maupun gas. Sensor suhu non-kontak memungkinkan sensor untuk mengukur suhu tanpa harus melakukan kontak fisik dengan objek yang akan diukur suhunya. Salah satu contoh sensor suhu non-kontak adalah sensor suhu yang bekerja menggunakan *thermal vision*.

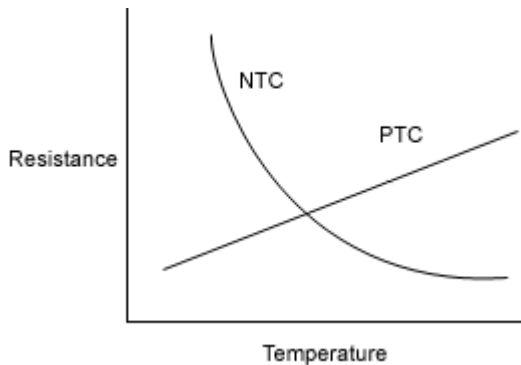
Sensor suhu terdiri dari beberapa jenis berdasarkan cara kerjanya. Sensor suhu yang pertama adalah *thermostat*. *Thermostat* berfungsi sebagai sensor suhu dengan tujuan saklar pemutus. *Thermostat* terdiri dari dua buah lempeng besi berbeda yang ditempelkan. Ketika kedua lempeng besi dipanaskan, maka masing-masing lempeng besi akan memuai dengan panjang yang berbeda sehingga menyebabkan lempeng besi menjadi bengkok. Bengkoknya lempeng besi mengakibatkan saklar akan terputus. Sensor suhu yang kedua adalah *thermistor*. *Thermistor* adalah sensor yang nilai resistansinya bergantung terhadap suhu. Sensor. Sensor suhu yang ketiga adalah *thermocouple*. *Thermocouple* bekerja dengan prinsip yang mirip dengan *thermostat*. Hanya saja *thermocouple* menjaga salah satu lempengnya pada suhu tertentu sebagai referensi. Perbedaan suhu antara kedua lempeng ini nantinya akan menghasilkan perbedaan potensial tegangan.

### 2.2.4 Thermistor

*Thermistor* adalah sebuah sensor suhu yang memiliki nilai resistansi yang bergantung pada suhu. *Thermistor* sensidri merupakan singkatan dari *Thermal Resistor*. *Thermistor* terdiri dari dua jenis yaitu PTC(*Positive Temperature Coefficient*) dan NTC(*Negative Temperature Coefficient*). *Thermistor* jenis PTC memiliki nilai resistansi yang berbanding lurus dengan kenaikan suhu. Artinya nilai resistansi dari PTC akan naik jika suhu yang diukur naik. *Thermistor* jenis NTC

memiliki nilai resistensi yang berbanding terbalik dengan kenaikan suhu. Artinya nilai resistansi dari NTC akan turun jika suhu yang diukur naik.

Kemampuan pengukuran suhu oleh *thermistor* berbeda-beda sesuai dengan produsen dari *thermistor*. Namun pada umumnya kemampuan *thermistor* berkisar antara  $-90^{\circ}\text{C}$  sampai  $130^{\circ}\text{C}$ . Penggunaan *thermistor* dalam kehidupan sehari-hari banyak dijumpai di detektor kebakaran, sensor suhu untuk kulkas, sensor suhu untuk *charging* baterai, sensor suhu untuk inkubator, dan masih banyak lagi yang berkaitan dengan pengukuran suhu.



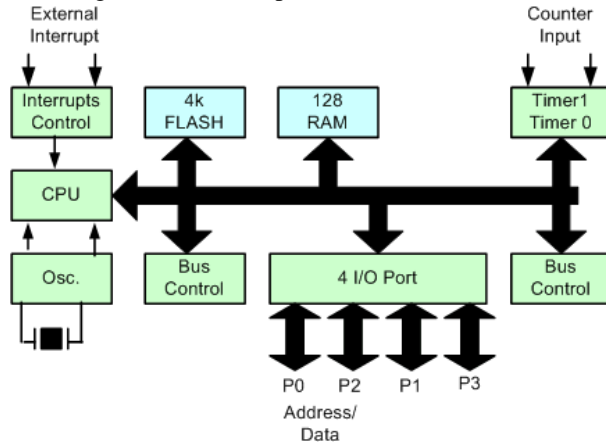
**Gambar 2.3** Ilustrasi Perubahan Resistansi PTC dan NTC[8]

### 2.2.5 Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer dalam satu *chip*. Mikrokontroler memiliki fungsi sebagai otak dari sebuah sistem elektronik. Perbedaan antaran mikrokontroler dan mikroprosesor adalah mikrokontroler sudah dilengkapi berbagai macam tambahan dalam *chip* tersebut. Mikrokontroler sudah dilengkapi dengan memori berupa ROM(*Read Only Memory*) dan RAM(*Random Acces Memory*). Sedangkan mikroprosesor tidak memiliki memori sehingga dibutuhkan memori eksternal. Mikrokontroler juga sudah dilengkapi dengan *port input/output* yang termasuk di dalam *chip*. Selain memori dan *port*, mikrokontroler juga sudah dilengkapi dengan ADC(*Analog-to-Digital Converter*), komunikasi serial, dan pewaktu/pencacah.

Kelebihan dari mikrokontroler adalah bentuknya yang kecil dan fiturnya yang lengkap memungkinkan mikrokontroler menjadi otak dari

sebuah sitem elektronika seperti robot, sistem alarm, sistem automasi, dan perangkat elektronik lainnya. Mikrokontroler juga lebih ekonomis dinandingkan dengan sistem komputer yang menggunakan modul-modul terpisah. Mikrokontroler juga memerlukan daya yang lebih rendah dibandingkan sistem komputer besar.



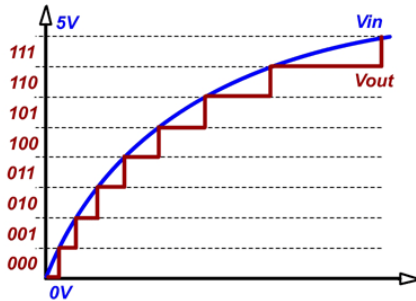
**Gambar 2.4** Arsitektur Mikrokontroler MCS51[9]

### 2.2.6 ADC

ADC adalah sebutan untuk rangkaian pengubah input sinyal analog (sinyal kontinu terhadap waktu) menjadi output sinyal digital (sinyal diskret atau terkuantisasi terhadap waktu). Seperti diketahui, komputer hanya bisa membaca sinyal diskret/biner sementara di dunia nyata segala sesuatunya secara fisis berupa kuantitas analog (suhu, tekanan, kecepatan, kelembapan, dsb). Kuantitas analog ini diubah menjadi besaran listrik (nilai tegangan atau arus yang setara) menggunakan transducer sebelum masuk rangkaian ADC untuk diubah menjadi sinyal digital. Sinyal digital inilah yang akan dibaca dan diproses oleh komputer.

Sinyal digital yang dihasilkan ADC berupa bilangan biner. Idealnya output sinyal tersebut harus dapat merepresentasikan kuantitas sinyal analog yang diterjemahkannya. Representasi ini akan semakin baik ketika ADC semakin sensitif terhadap perubahan nilai sinyal analog yang masuk. Sensitifitas dari ADC biasanya disebut resolusi. Resolusi

adalah jumlah bit output pada ADC. Sebuah rentang sinyal analog dapat dinyatakan dalam kode bilangan digital sehingga menyatakan sebuah sinyal analog dalam rentang 16 skala (4 bit) adalah lebih baik resolusinya dibanding membaginya dalam rentang 8 skala (3 bit). Karena besar resolusi sebanding dengan  $2^N$  ( $N$  adalah jumlah bit output digital pada rangkaian pengubah) semakin besar jumlah bit, resolusi akan semakin baik[10].



**Gambar 2.5** Ilustrasi I/O ADC 3bit[11]

### 2.2.7 Ethernet

Ethernet adalah teknologi jaringan komputer yang termasuk ke dalam jaringan dengan lingkup lokal atau biasa disebut LAN(*Local Area Network*). Arsitektur Ethernet ini di perkenalkan tahun 1970-an oleh Xerox. Ethernet memiliki standar IEEE 802.3. Terdapat tiga jenis Ethernet yang di bedakan berdasarkan kecepatan akses datanya. Jenis yang pertama adalah Ethernet. Ethernet memiliki kecepatan transfer data hingga 10 Mbps. Jenis yang kedua adalah *Fast Ethernet*. *Fast Ethernet* memiliki kecepatan transfer data hingga 100 Mbps. Jenis yang ketiga adalah Gigabit Ethernet. Gigabit Ethernet memiliki kecepatan transfer data hingga 1000 Mbps atau setara dengan 1 Gbps.

Arsitektur ini memiliki kemampuan berbagi *bandwidth* atau *resource* dalam suatu jaringan yang sama. Protokol yang di gunakan pada arsitektur ini adalah CSMA/CD atau *Carrier Sense Multiple Access With Collision Detections*, protokol ini berfungsi supaya tidak terjadi *collusion* ketika berbagi *bandwidth*. *Collusion* merupakan suatu kondisi di mana ada beberapa *source computer* yang mengirimkan data pada saat yang sama sehingga data yang dikirimkan melalui jaringan tersebut akan bertabrakan dan data tidak akan terkirim.

Sebelumnya, Ethernet utamanya digunakan dalam teknologi akses, menyediakan akses internet atau *interface user* ke jaringan. Sampai saat ini kondisi tersebut masih berjalan tetapi standar ethernet sendiri dikembangkan untuk mampu melayani layanan data pada jaringan *transport*. Fungsi-fungsi layanan pada teknologi Ethernet sebagai jaringan *transport* merupakan hasil pengembangan yang terus-menerus[12].

### **2.2.8 Protokol Modbus**

Modbus adalah protokol yang umum digunakan oleh berbagai perangkat untuk komunikasi sederhana, handal, dan efisien di berbagai jaringan modern. Protokol industri modbus dikembangkan pada tahun 1979 untuk membuat komunikasi yang memungkinkan antara perangkat otomatisasi. Awalnya diimplementasikan sebagai protokol level aplikasi yang dimaksudkan untuk mentransfer data melalui serial, kemudian protokol telah diperluas untuk mencakup implementasi selain serial yaitu TCP / IP, dan User Datagram Protocol (UDP).

Modbus biasanya digunakan untuk jaringan komunikasi antar divais dalam Supervisory Control dan Data Acquisition (SCADA). Sebagai contoh, server besar dapat digunakan sebagai master untuk Programmable Logic Controller (PLC) atau Programmable Automation Controller (PAC), sedangkan PLC / PAC menjadi master bagi sensor, katup, motorik, atau divais embedded lainnya.

Untuk memenuhi kebutuhan untuk jaringan komunikasi antar divais dalam SCADA, Modbus dirancang sebagai protokol request-response dengan data dan model fungsi yang fleksibel[13].

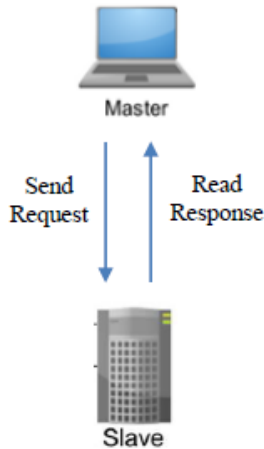
Protokol Modbus mengikuti arsitektur master dan slave dimana master mengirimkan permintaan (request) untuk slave dan menunggu balasan (respon). Arsitektur ini memberikan master kontrol penuh atas arus informasi, yang memiliki manfaat pada jaringan serial older multidrop. Bahkan pada jaringan TCP / IP modern, memberikan master kontrol tinggi atas slave, yang memberikan keuntungan dalam beberapa desain.

Dalam Modbus, request adalah data dengan beberapa layer . Lapisan pertama adalah Application Data Unit (ADU), yaitu yang sering dianggap sebagai "jenis" dari Modbus yang digunakan. Ada tiga ADU: ASCII, Remote Terminal Unit (RTU), dan TCP / IP. TCP adalah format modern yang memungkinkan untuk penanganan efisien untuk request Modbus dan respon dalam perangkat lunak, serta jaringan yang lebih

efisien melalui penggunaan koneksi berdedikasi dan pengidentifikasi untuk setiap request. RTU dan ASCII adalah format ADU serial yang lebih tua dengan perbedaan utama antara kedua adalah bahwa RTU menggunakan representasi biner compact sementara ASCII mengirimkan semua request sebagai aliran karakter ASCII.

Modbus mengelola akses data secara sederhana dan fleksibel. Modbus mendukung dua tipe data: nilai Boolean dan unsigned, 16-bit integer. Dalam sistem SCADA, umumnya divais embedded memiliki nilai-nilai tertentu yang didefinisikan sebagai input, seperti pengaturan proporsional atau Proporsional Integral Derivatif (PID), sementara nilai-nilai lain adalah output, seperti suhu atau posisi katup. Untuk memenuhi kebutuhan ini, nilai data Modbus dibagi menjadi empat range, coil, input diskrit, holding register dan input register. Slave dapat menentukan sebanyak 65.536 elemen dalam setiap range.

Modbus mengekspos nilai-nilai ini dengan cara yang benar-benar konseptual, yang berarti mungkin sebenarnya tidak ada dalam memori. Sebagai contoh, divais slave dapat didefinisikan sehingga holding register dan input register benar-benar berbagi memori yang sama jika memungkinkan bagi slave. Dalam kebanyakan kasus, slave menyimpan setiap jenis data yang mendukung dalam memori terpisah, dan membatasi jumlah elemen data yang dapat diakses oleh master. Fleksibilitas ini adalah pilihan karena dengan cara demikian data yang terekspos dapat didefinisikan dengan kode fungsi Modbus. Kode fungsi Modbus menentukan bagaimana data diakses dan dimodifikasi oleh master[13].



**Gambar 2.6** Hubungan *Request-Response* pada Divais ModBus[13]

### 2.2.9 HMI

HMI adalah singkatan dari *Human Machine Interface*. HMI secara umum merepresentasikan sebuah *interface* atau tampilan penghubung antara manusia dengan mesin. HMI merupakan tempat dimana user melakukan pengawasan atau monitoring pada proses yang ada di sistem. Selain itu user juga dapat memasukkan input pada tampilan HMI[14].

### 2.2.10 Kalor

Kalor adalah salah satu bentuk energi yang berpindah karena adanya perbedaan suhu. Kalor berpindah dari suhu yang lebih tinggi ke suhu yang lebih rendah. Kalor dapat menaikkan atau menurunkan suhu. Ketika suatu benda mengalami kenaikan suhu maka benda tersebut sedang menyerap kalor. Sedangkan ketika suatu benda mengalami penurunan suhu maka benda tersebut sedang melepas kalor.

Semakin besar kenaikan suhu maka kalor yang diterima akan semakin besar. Semakin kecil kenaikan suhu maka kalor yang diterima akan semakin kecil. Maka hubungan kalor berbanding lurus atau sebanding dengan kenaikan suhu jika massa dan jenis tetap. Semakin besar massa zat maka kalor yang diperlukan akan semakin besar. Semakin kecil massa zat maka kalor yang diperlukan akan

semakin kecil. Kalor jenis zat adalah banyaknya kalor yang dibutuhkan untuk menaikkan suhu 1 kg zat sebesar 1<sup>0</sup>C. Kalor memiliki persamaan yang diuraikan dalam Persamaan 2.1.

$$Q = m \cdot c \cdot \Delta T \quad (2.1)$$

Q = kalor (J)

c = kalor jenis (J/kg<sup>0</sup>C)

m = massa (kg)

ΔT = perubahan suhu (<sup>0</sup>C)

### 2.2.11 AC

AC adalah singkatan dari *Air Conditioner* yang jika diartikan kedalam bahasa indonesia adalah pengondisi udara. AC lebih sering disebut sebagai pendingin ruangan dalam bahasa indonesia. AC berfungsi untuk mendinginkan ruangan sesuai dengan suhu ruangan yang diinginkan. AC mendinginkan ruangan dengan memanfaatkan *refrigent* yang menyerap atau melepaskan kalor saat *refrigent* berubah tekanan dan fase. AC memiliki komponen-komponen utama untuk merubah tekanan dan fase *refrigent* berupa kompresor, kondensor dan evaporator. Kompresor berfungsi untuk menaikkan tekanan *refrigent* yang berfase gas menjadi gas bertekanan tinggi yang akan dialirkan ke kondensor. Kondensor berada dalam unit luar ruangan yang berfungsi merubah fase *refrigent* gas menjadi fase cair bertekanan tinggi yang akan dialirkan ke evaporator. Saat *refrigent* berubah fasa dalam kondensor maka *refrigent* akan melepaskan kalor. Evaporator berfungsi untuk merubah *refrigent* cair dari kondensor menjadi *refrigent* fase gas. Saat *refrigent* berubah fase dari cair ke gas maka *refrigent* akan menyerap kalor sehingga menjadi dingin. *Refrigent* dingin ini akan mengalir dalam kumparan pendingin yang berada di unit dalam dan kumparan pendingin akan dilewati udara melalui kisi-kisinya yang didorong oleh kipas untuk memutar udara dalam ruangan sehingga menghasilkan udara yang dingin.

Dalam unit dalam terdapat filter yang berfungsi menyaring kotoran udara agar tidak masuk ke unit AC. Jika filter dalam kondisi kotor maka sirkulasi udara akan terhambat dan juga kisi-kisi dalam unit dalam kotor yang akan menghambat pertemuan udara yang dipompa dengan kumparan pendingin sehingga kerja AC tidak maksimal.

Kapasitas pendinginan AC biasanya tercantum dalam BTU. BTU adalah singkatan dari *British Thermal Unit*. BTU adalah perhitungan



kebutuhan kapasitas pendinginan AC yang optimal terhadap ukuran ruangan. Perhitungan BTU dapat dilakukan dengan menggunakan Persamaan 2.2.

$$BTU/H = (W.H.I.L.E)/60 \quad (2.2)$$

W = lebar ruangan (kaki)

H = tinggi ruangan (kaki)

I = 10 jika berinsulasi, 18 jika tidak berinsulasi

L = panjang ruangan (kaki)

E = 16 jika dinding terpanjang menghadap utara, 17 menghadap timur,  
18 menghadap selatan, 20 menghadap barat

*# Halaman ini sengaja dikosongkan #*

## **BAB 3**

### **PERANCANGAN SISTEM**

Pada bab perancangan sistem akan dijelaskan rancangan dari sistem *monitoring* waktu pendinginan ruangan untuk deteksi kebersihan filter AC. Rancangan sistem mengandung penjelasan mengenai diagram blok sistem, desain perangkat keras, dan perangkat lunak sistem.

#### **3.1 Diagram Blok Sistem**

Diagram blok sistem menjelaskan peran setiap blok-blok dari sistem dan cara kerjanya. Sistem memiliki beberapa blok diantaranya AC, PLN, sensor arus, mikrokontroler, sensor suhu, dan HMI.

AC(*Air Conditioner*) adalah pendingin ruangan yang berfungsi untuk mendinginkan ruangan sesuai dengan suhu ruangan yang diinginkan. AC mendinginkan ruangan dengan memanfaatkan *refrigent* yang menyerap atau melepaskan kalor saat *refrigent* berubah tekanan dan fase.

PLN dalam sistem ini merujuk pada sumber listrik untuk unit AC. PLN menghasilkan listrik sebesar 220 VAC. Ketika AC bekerja maka AC akan mengkonsumsi arus dari PLN.

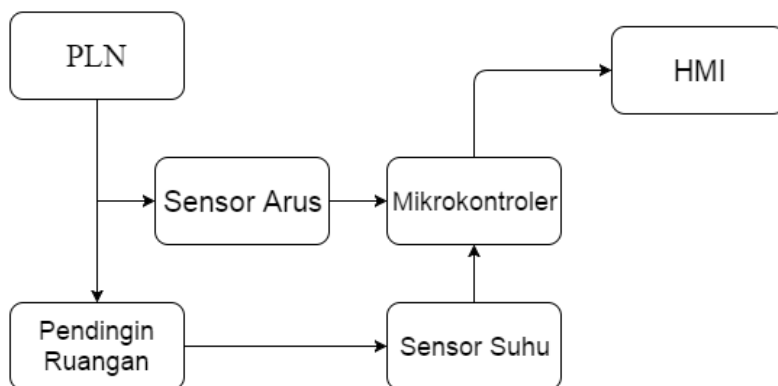
Sensor arus merupakan alat yang berfungsi untuk mengukur arus yang digunakan oleh sebuah beban listrik. Sensor arus akan menghasilkan output tegangan yang menunjukkan jumlah arus yang digunakan oleh beban. Dalam sistem ini sensor arus akan mengukur jumlah arus yang dipakai oleh AC dan menghasilkan tegangan yang akan dibaca oleh mikrokontroler.

Sensor suhu merupakan alat yang berfungsi merubah besaran suhu menjadi besaran listrik. Sensor suhu akan menghasilkan output tegangan yang menunjukkan besarnya suhu yang diukur. Dalam sistem ini sensor suhu akan mengukur suhu ruangan dan output tegangannya akan dibaca oleh mikrokontroler.

Mikrokontroler adalah sebuah chip yang dapat mengatur rangkaian elektronik dan dapat membaca input data dari rangkaian elektronik. Mikrokontroler dapat menyimpan program untuk dapat mengontrol rangkaian elektronik. Dalam sistem ini mikrokontroler berfungsi untuk membaca data sensor arus dan sensor suhu, mengolah data sensor, dan mengirim data ke HMI. Mikrokontroler mengolah data arus dan sensor untuk menghitung daya yang dikonsumsi AC dan suhu

ruangan dan menentukan apakah filter AC dalam keadaan bersih atau kotor. Mikrokontroler juga akan mengirim data ke HMI untuk menampilkan keadaan filter AC. Selain itu mikrokontroler juga akan memberikan indikator berupa lampu LED yang berwarna hijau ketika filter bersih dan berwarna merah ketika filter kotor.

HMI(*Human Machine Interface*) adalah perangkat yang menjembatani komunikasi antara manusia dan mesin. Manusia dan mesin memiliki bahasa yang berbeda dan HMI memungkinkan interaksi antara manusia dan mesin dapat dilakukan. Dalam sistem ini HMI berfungsi untuk menampilkan kondisi filter AC, dan juga menampilkan parameter-parameter lain yang diperlukan sebagai informasi oleh manusia. Diagram blok dari sistem digambarkan dalam Gambar 3.1.



**Gambar 3.1** Diagram Blok Sistem

### 3.2 Perancangan Perangkat Keras(*Hardware*)

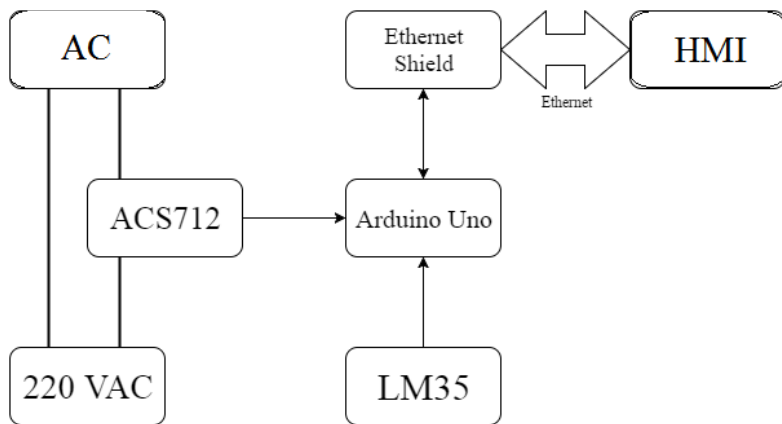
Diagram blok sistem yang digambarkan pada Gambar 3.1 diuraikan lagi setiap bloknya sehingga menjadi diagram blok *hardware* yang lebih spesifik.

AC yang digunakan dalam sistem ini adalah haier HSU-05LEA03, Sharp AH-AP5MHL, dan Midea MPG-09CR. Sensor arus yang digunakan adalah ACS712 yang dipasang secara seri dengan AC. Untuk mengukur daya nantinya hasil dari pengukuran sensor arus akan dikalikan dengan 220 dikarenakan kebutuhan pengukuran daya hanya sebagai penentu apakah AC sedang bekerja atau tidak. Sensor suhu yang

digunakan dalam sistem ini adalah LM35. Baik sensor arus maupun sensor suhu menggunakan *supply* 5V dari mikrokontroler. Mikrokontroler yang digunakan dalam sistem ini adalah Arduino Uno. HMI yang digunakan dalam sistem ini adalah simulasi komputer.

Komunikasi antara mikrokontroler dan HMI menggunakan ethernet. Untuk itu digunakan ethernet shield agar Arduino dapat menggunakan ethernet.

Diagram blok *hardware* lebih spesifik digambarkan dalam Gambar 3.2.



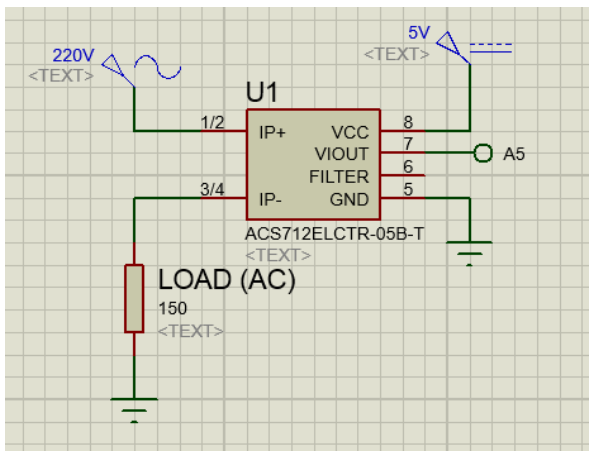
**Gambar 3.2** Diagram Blok *Hardware*

### 3.2.1 Perancangan Sensor Arus

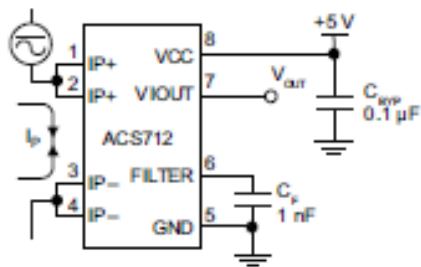
Sensor arus menggunakan IC ACS712 yang sudah berbentuk modul. ACS712 merupakan suatu IC terpakat yang mana berguna untuk sensor arus menggantikan trafo arus yang relatif besar dalam hal ukuran. Pada prinsipnya ACS712 sama dengan sensor efek hall lainnya yaitu dengan memanfaatkan medan magnetik di sekitar arus kemudian dikonversi menjadi tegangan yang linier dengan perubahan arus. Nilai variabel dari sensor ini merupakan input untuk mikrokontroler yang kemudian diolah[15].

*Supply* dari PLN akan dihubungkan dengan input IP+ dan output IP- akan dihubungkan dengan AC. ACS712 menggunakan *supply* 5v dari Arduino. Output sinyal dari ACS712 akan dihubungkan dengan port

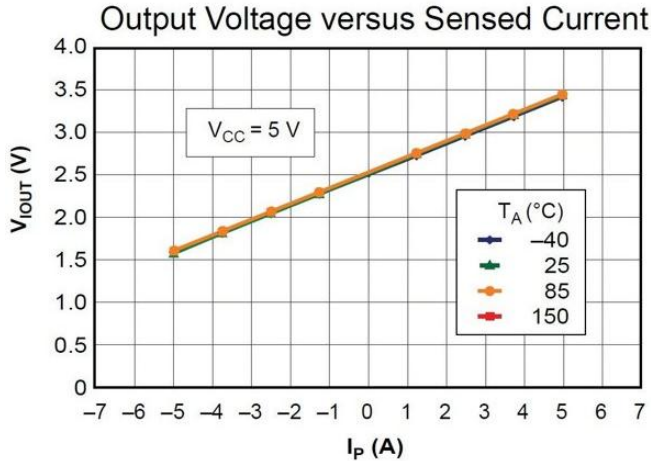
A5 dari Arduino. Arduino Uno memiliki ADC 10bit dengan tegangan referensi *default*nya 5 V sehingga memiliki ketelitian 0,0048 V. ACS712 memiliki ketelitian 0,185 V/A. Pengukuran arus pada sistem memerlukan pembacaan hingga ketelitian 0,1 A. Ketelitian ADC Arduino Uno lebih kecil dibandingkan ketelitian ACS712, maka ADC dari Arduino Uno akan mampu membaca *output* dari ACS712. Rangkaian sensor arus digambarkan pada Gambar 3.3.



**Gambar 3.3** Rangkaian Sensor Arus



**Gambar 3.4** Rangkaian Umum ACS712[16]

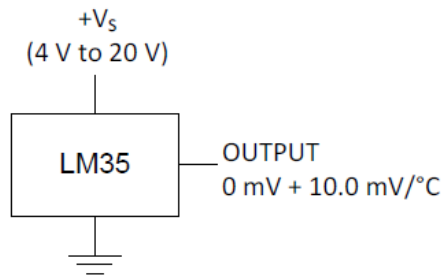


**Gambar 3.5** Karakteristik Input Output ACS712[16]

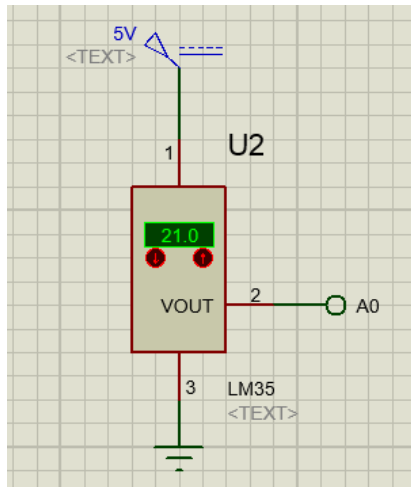
### 3.2.2 Perancangan Sensor Suhu

Sensor suhu yang digunakan dalam sistem ini adalah LM35. Sensor LM35 adalah sebuah perangkat yang mengubah besaran suhu menjadi besaran listrik. Besaran listrik yang dihasilkan oleh LM35 linier dengan perubahan suhu. LM35 dapat mengukur suhu dari  $2^\circ\text{C}$  sampai  $150^\circ\text{C}$  jika digunakan dalam pengukuran suhu standar. LM35 dapat menggunakan supply antara 4V sampai 20V[17]. Dalam sistem ini LM35 menggunakan *supply* 5v dari Arduino.

LM35 mempunyai jangkauan temperatur antara  $0$ - $100^\circ\text{C}$ . Setiap perubahan  $1^\circ\text{C}$  maka output tegangan LM35 akan berubah sebesar 10mV. Output dari LM35 akan dihubungkan dengan port A0 dari Arduino Uno. Arduino Uno memiliki ADC 10bit dengan tegangan referensi *default*nya 5 V sehingga memiliki ketelitian 0,0048 V. LM35 memiliki ketelitian  $0,01\text{ V}/^\circ\text{C}$ . Ketelitian ADC Arduino Uno lebih kecil dibandingkan ketelitian LM35, maka ADC dari Arduino Uno akan mampu membaca *output* dari LM35. Rangkaian sensor suhu digambarkan dalam Gambar 3.7.



**Gambar 3.6** Basic Centigrade Temperature Sensor[17]



**Gambar 3.7** Rangkaian Sensor Suhu

### 3.2.3 Rangkaian Mikrokontroler

Mikrokontroler yang digunakan dalam sistem ini adalah Arduino Uno. Arduino merupakan platform dari physical computing. Istilah platform ini dikarenakan Arduino tidak hanya merupakan sebuah alat pengembangan, tetapi juga perpaduan antara hardware, bahasa pemrograman, dan Integrated Development Environment (IDE) yang bersifat open source (baik untuk hardware maupun software-nya)[13].



Kemudian, suatu physical computing digunakan untuk membuat sebuah sistem yang bersifat interaktif, yaitu dapat menerima maupun memberikan respon terhadap rangsangan dari lingkungan. Artinya, sistem dibuat dengan konsep menghubungkan lingkungan yang secara alami bersifat analog dengan dunia digital (komputer). Hal ini dapat diwujudkan dengan menggunakan sensor dan aktuator serta mikrokontroler[13].

Saat ini, banyak desain dan alat yang dikembangkan oleh akademisi dan profesional dengan menggunakan Arduino. Berbagai pihak telah membuat berbagai modul pendukung (seperti sensor, display, penggerak, dan lain sebagainya) untuk dapat melengkapi dan disambungkan dengan Arduino. Arduino menjadi pilihan dan acuan bagi banyak praktisi sehingga Arduino telah berevolusi menjadi sebuah platform. Alasan dari banyaknya pengembangan dari Arduino antara lain murah, lintas platform, mudah dipelajari dan digunakan, serta open source[13].

Secara umum, Arduino terdiri atas hardware dan software. Hardware Arduino adalah board dengan beberapa input/output (I/O), sedangkan software Arduino mencakup IDE untuk menuliskan kode program atau sketch, driver untuk koneksi antara Arduino dengan komputer, beberapa contoh program (example), dan library untuk pengembangan program[13].

IDE Arduino adalah software yang ditulis menggunakan Java. IDE Arduino terdiri atas editor program, compiler, dan uploader. Editor program merupakan sebuah window untuk menuliskan dan melakukan editing kode program atau sketch untuk mikrokontroler Arduino dalam bahasa Processing. Setelah itu, diperlukan compiler sebagai modul untuk mengubah kode program (dalam bahasa Processing) menjadi kode biner sehingga dapat dipahami mikrokontroler. Hal ini dikarenakan sebuah mikrokontroler tidak dapat memahami bahasa processing. Selain itu, dengan menggunakan compiler, dapat dilihat apakah kode program telah ditulis dengan benar (tidak ada kesalahan atau error). Kemudian, digunakan uploader, yakni modul yang memuat kode biner setelah di-compile dari komputer ke dalam memori yang terdapat dalam board Arduino. Pada IDE Arduino dapat dipilih jenis Arduino yang digunakan sehingga upload dapat dilakukan[13].

Adapun spesifikasi board arduino uno adalah sebagai berikut[18]:

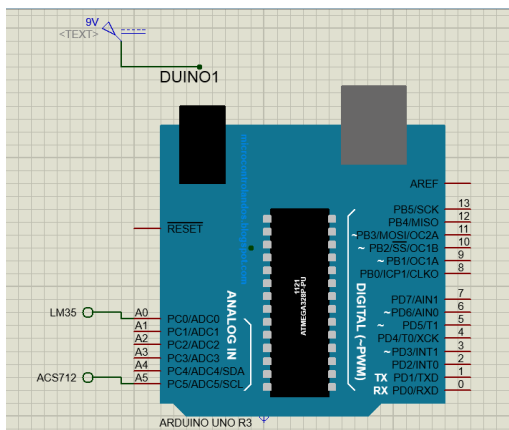
- Mikrokontroler: Atmega328P
- Tegangan Operasi: 5V

- Tegangan Input (recommended) : 7 - 12 V
- Tegangan Input (limit) : 6-20 V
- Pin digital I/O : 14 (6 diantaranya pin PWM)
- Pin Analog input : 6
- Arus DC per pin I/O : 20 mA
- Arus DC untuk pin 3.3 V : 50 mA
- Flash Memory : 32 KB dengan 0.5 KB digunakan untuk bootloader
- SRAM : 2 KB
- EEPROM : 1 KB
- Kecepatan Pewaktuan : 16 Mhz

Arduino Uno dalam sistem ini menggunakan *supply* dari adaptor 9v. Output 5v dari Arduino Uno digunakan sebagai *supply* untuk ACS712 dan LM35.

Sistem ini menggunakan ADC dari Arduino Uno untuk membaca output dari sensor suhu dan sensor arus. Output dari sensor arus ACS712 dihubungkan dengan port A0 dari Arduino Uno. Output dari sensor suhu LM35 dihubungkan dengan port A5 dari Arduino Uno.

Arduino Uno berkomunikasi dengan HMI menggunakan protokol ModBus melalui kabel ethernet. Sehingga diperlukan ethernet shield agar Arduino dapat berkomunikasi menggunakan ethernet. Rangkaian mikrokontroler digambarkan dalam Gambar 3.8.



**Gambar 3.8** Rangkaian Mikrokontroler

### 3.2.4 Rangkaian HMI

HMI yang digunakan dalam sistem ini adalah simulasi *touchscreen* Weintek MT8070iH. Simulasi dijalankan pada komputer menggunakan *Software* EasyBuilder8000. Mikrokontroler berkomunikasi dengan komputer menggunakan kabel ethernet melalui Ethernet Shield.

Ketika Arduino Uno menggunakan Ethernet Shield akan ada beberapa *port* yang terpakai oleh Ethernet Shield sehingga tidak bisa digunakan lagi. Pin *digital* 4 digunakan oleh Ethernet Shield untuk memilih SD Card, pin *digital* 10 digunakan untuk memilih W5100, dan pin *digital* 11 sampai 13 digunakan oleh Bus SPI[19].

Karena W5100 dan SD card berbagi bus SPI, hanya salah satu yang dapat aktif pada satu waktu. Jika kita menggunakan kedua perangkat dalam program kita, hal ini akan diatasi oleh library yang sesuai. Jika kita tidak menggunakan salah satu perangkat dalam program kita, kiranya kita perlu secara eksplisit mendeselect-nya. Untuk melakukan hal ini pada SD card, set pin 4 sebagai output dan menuliskan logika tinggi padanya, sedangkan untuk W5100 yang digunakan adalah pin 10. DFRduino Ethernet shield adalah sebuah clone dari arduino Ethernet shield yang dibuat oleh DFRobot[19].

Diagram HMI digambarkan pada Gambar 3.9.



**Gambar 3.9** Diagram Blok HMI

### 3.2.5 Perangkat yang Dimonitor

Pada Tugas Akhir ini dilakukan percobaan terhadap 3 AC. AC yang pertama dan kedua adalah Haier HSU-05LEA03 dan Sharp AH-AP5MHL. Kedua AC tersebut adalah AC dengan jenis sama yaitu AC *Split* 0,5 pk. Akan tetapi kedua AC tersebut berada di dalam ruangan dengan ukuran yang berbeda. AC Haier HSU-05LEA03 berada dalam ruangan 2,5x2,5x2,5 meter. AC Sharp AH-AP5MHL berada dalam ruangan yang lebih besar 3x4x3 meter. Tujuan digunakan dua AC yang berbeda tersebut adalah untuk membandingkan kemampuan alat untuk mendeteksi AC dengan beban yang berbeda. AC yang ketiga adalah AC *portable* Midea MPG-09CR. Tujuan digunakannya AC *portable* adalah

untuk kepentingan demo alat saat sidang karena tidak memungkinkan untuk melakukan demo pada ruangan AC dimana pengujian alat penulis lakukan.

Spesifikasi dari AC haier HSU-05LEA03 adalah:

- Tipe : Split
- Daya : 330 Watt
- Arus : 1.5 A
- 0,5 pk



**Gambar 3.10** Haier HSU-05LEA03[20]

Spesifikasi dari AC Sharp AH-AP5MHL adalah:

- Tipe : Split
- Daya : 330 Watt
- Arus : 1.6 A
- 0,5 pk



**Gambar 3.11** Sharp AH-AP5MHL[21]

Spesifikasi dari AC Midea MPG-09CR adalah:

- Kapasitas pendinginan : 9000 Btu
- Daya : 900 W
- Ukuran *Indoor* : 433x355x840 mm
- Berat bersih : 30 KG



**Gambar 3.12** Midea MPG-09CR[22]

### **3.3 Perancangan Perangkat Lunak(*Software*)**

Perancangan perangkat lunak dilakukan dalam dua tahap, yaitu perancangan perangkat lunak mikrokontroler dan perancangan *interface* HMI. Perancangan perangkat lunak mikrokontroler dilakukan dengan menggunakan IDE Arduino. Perancangan *interface* HMI dilakukan dengan menggunakan EasyBuilder8000.

#### **3.3.1 Perancangan Program Mikrokontroler**

Perancangan program Arduino menggunakan *software* IDE Arduino. Dalam program ini diperlukan *library* ModBus, ethernet dan emon. *Library* ModBus diperlukan karena komunikasi antara Arduino dengan HMI menggunakan protokol ModBus. *Library* ethernet digunakan karena Arduino dan HMI dihubungkan menggunakan kabel ethernet. *Library* emon digunakan untuk mengukur arus dari ACS712.

Dalam *library* ModBus, terdapat fungsi untuk membaca konten biner dari *holding register* (referensi 4X) dan preset nilai-nilai secara berurutan dengan *holding register* (referensi 4X). Alamat register digunakan untuk data suhu, kecepatan pendinginan AC, estimasi daya

yang terbuang, dan kondisi filter AC. Penggunaan ModBus dan Ethernet dalam Sistem adalah sebagai berikut:

```
#include <Mudbus.h>
#include <SPI.h>
#include <Ethernet.h>
Mudbus Mb;

uint8_t mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x51, 0x06 };
uint8_t ip[] = { 192, 168, 1, 8 };
uint8_t gateway[] = { 192, 168, 1, 1 };
uint8_t subnet[] = { 255, 255, 255, 0 };
Ethernet.begin(mac, ip, gateway, subnet);
//Avoid pins 4,10,11,12,13 when using ethernet shield
delay(5000); //Time to open the terminal

Mb.Run();
Mb.R[0]=cel;
Mb.R[3]=kondisi;
```

Mikrokontroler akan menyimpan nilai suhu dan kondisi AC dalam register yang kemudian dikirimkan ke HMI melalui Ethernet. HMI kemudian akan membaca nilai suhu dari register Mb.R[0], dan kebersihan filter dari register Mb.R[3]. Nilai suhu dan kondisi kebersihan filter AC ini nantinya akan ditampilkan melalui *interface* HMI.

Mikrokontroler mengukur arus yang dikonsumsi AC menggunakan *library* emon. Dengan menggunakan *library* emon mikrokontroler dapat mengukur nilai arus RMS yang dikonsumsi oleh AC. Untuk meningkatkan kepresisian pengukuran arus, *library* emon sudah dilengkapi dengan opsi jumlah gelombang yang diukur untuk setiap iterasinya, dan besaran koefisien kalibrasi pengukuran. Program pengukuran arus dengan *library* emon oleh mikrokontroler adalah sebagai berikut:

```
#include "EmonLib.h"
EnergyMonitor emon1;

emon1.current(5, 10); // arus; input pin, kalibrasi
```

```
emon1.calcVI(10,200); // hitung Irms. Jumlah gelombang, time-out.
```

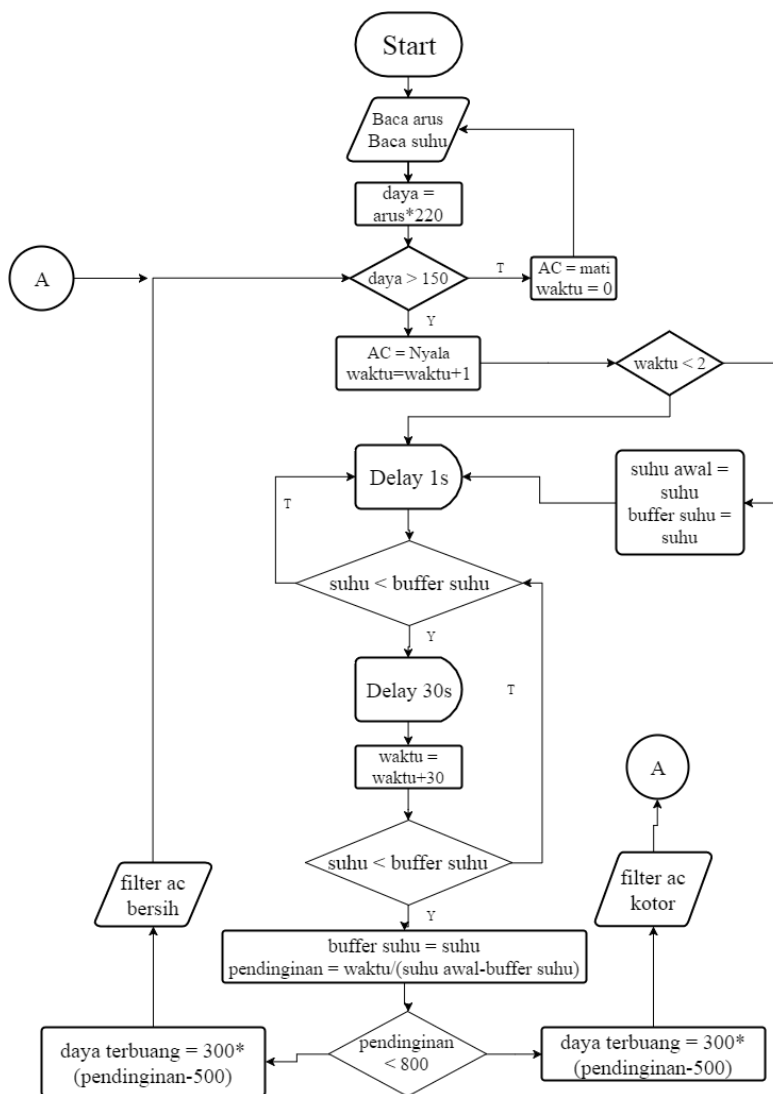
```
Float current = emon1.Irms; //ekstrak Irms kedalam variabel
```

Mikrokontroler mengukur arus menggunakan metode pembacaan ADC. Penggunaan ADC pada IDE Arduino untuk membaca nilai suhu terdiri dari dua tahap. Tahap pertama adalah membaca nilai ADC dari LM35. Ketika sudah didapatkan nilai ADC kemudian tahap kedua dilakukan penghitungan terhadap nilai tegangan referensi ADC dan karakter sinyal keluaran LM35 untuk mendapatkan suhu dalam satuan Celcius. Pengukuran suhu oleh mikrokontroler adalah sebagai berikut:

```
val = analogRead(tempPin); //membaca nilai ADC  
float mv = (val*5.0/1024.0)*5000; //menghitung tegangan analog  
float cel = mv/10; //menghitung temperatur dalam Celcius
```

Dari *coding* diatas, mikrokontroler membaca data arus melalui analog pin 5 dengan kalibrasi sebesar 10. Lalu *library* emon melakukan *sampling* sebanyak 10 gelombang dengan batas waktu 200ms. Selanjutnya nilai arus disimpan dalam variabel *current*. Nilai arus nantinya akan dikalikan dengan 220 untuk mendapatkan daya.

Mikrokontroler akan diprogram untuk menghitung daya untuk mengetahui apakah AC nyala atau mati. Jika AC nyala maka mikrokontroler akan menyimpan suhu awal. Selanjutnya jika suhu turun maka mikrokontroler akan mencatat suhu tersebut dan waktunya. Jika kecepatan pendinginan ( $s/^{\circ}C$ ) lebih besar dari *threshold* yang ditentukan maka AC sudah kotor.



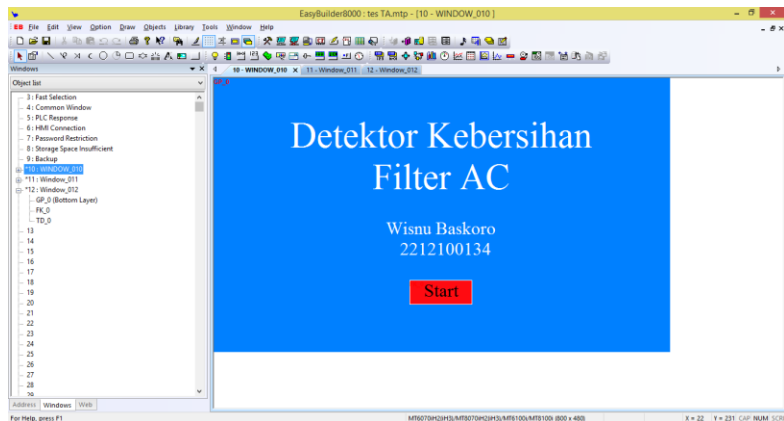
**Gambar 3.13** Flow Chart Program Mikrokontroler



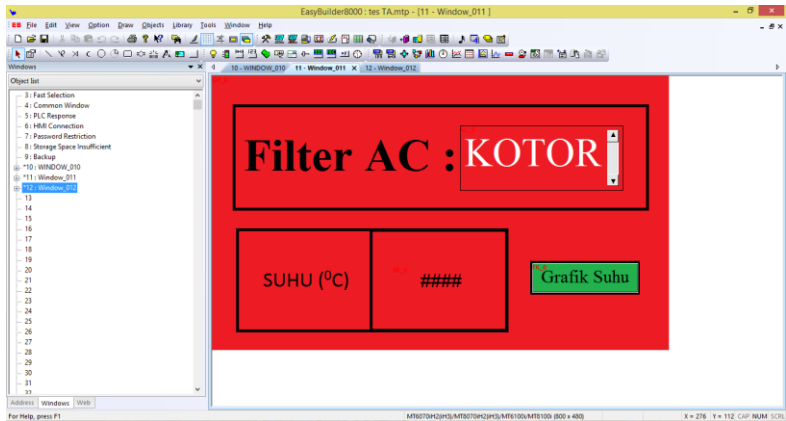
### 3.3.2 Pemrograman HMI

Pemrograman *interface* untuk simulasi Weintek MT8070iH menggunakan EasyBuilder8000. Untuk sistem deteksi kebersihan filter AC ini digunakan beberapa *object* untuk interface, antara lain:

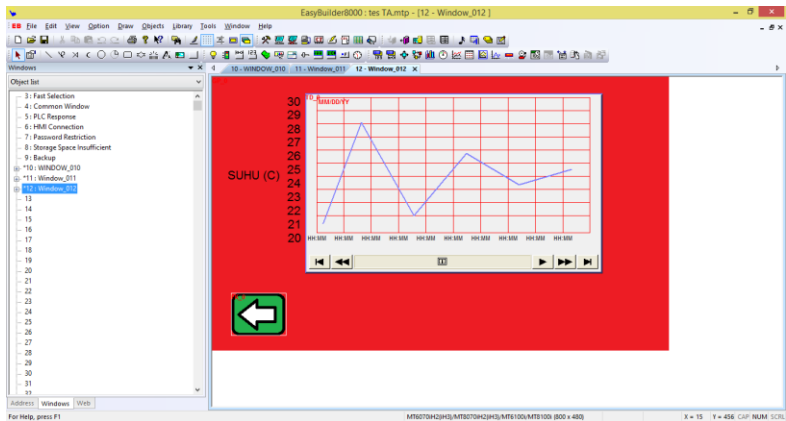
1. *Numeric Object* untuk menampilkan suhu ruangan. Alamat register yang digunakan untuk *Numeric Object* ini adalah Mb.R[0] pada program Arduino. Sedangkan untuk alamat yang dipilih pada pemrograman *interface* adalah 4X 1.
2. *Alarm Display* untuk menampilkan kondisi filter AC. Alamat register yang digunakan untuk *Alarm Display* ini adalah Mb.R[3] pada program Arduino. Sedangkan untuk alamat yang dipilih pada pemrograman *interface* adalah 4X 4.
3. *Trend Display* untuk menampilkan grafik suhu. Alamat register yang digunakan untuk *Trend Display* ini adalah Mb.R[0] pada program Arduino. Sedangkan untuk alamat yang dipilih pada pemrograman *interface* adalah 4X 1.



**Gambar 3.14** Pemrograman *Interface*



**Gambar 3.15** Pemrograman *Interface* (2)

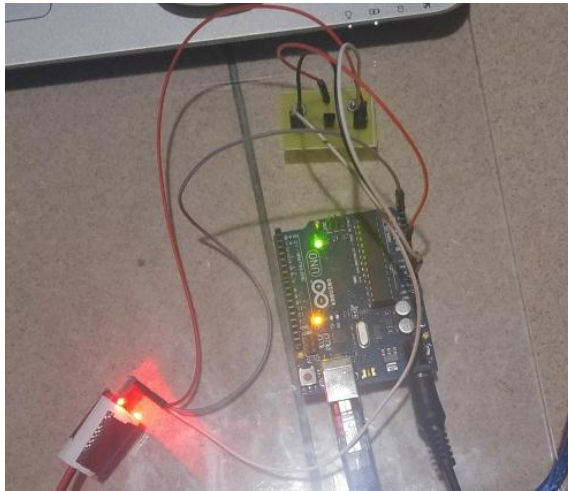


**Gambar 3.16** Pemrograman *Interface* (3)

## **BAB 4**

### **PENGUJIAN DAN ANALISIS**

Pada bab ini akan dipaparkan hasil pengambilan data dan pengujian sistem. Data yang akan dipaparkan berupa pengukuran daya, pengukuran suhu, pengujian sistem, dan simulasi *interface*.



**Gambar 4.1** Rangkaian Sistem



**Gambar 4.2** HMI Sistem



**Gambar 4.3** Haier HSU-05LEA03



**Gambar 4.4** Sharp AH-AP5MHL



**Gambar 4.5** Midea MPG-09CR



**Gambar 4.6** Ruangn Buatan

#### 4.1 Pengukuran Arus

Pengujian pengukuran arus menggunakan ACS712 bertujuan untuk mengetahui fungsi input output dari sensor arus sistem. Pengujian dilakukan dengan menggunakan *adjustable supply* yang dirangkai secara seri dengan beban, ACS712, dan amperemeter. *Supply* diubah-ubah dari arus 0.0 A sampai 2 A. Hasil pengukuran ditunjukkan pada Tabel 4.1 dan Gambar 4.7.

Dari tabel 4.1 dan Gambar 4.7 dapat terlihat adanya eror yang cukup besar walaupun sudah linear. Maka pasangan input output tadi diolah menggunakan persamaan polinomial derajat dua sehingga menghasilkan Persamaan 4.1:

$$I_r = -0.0374I_s^2 + 0.8912I_s - 0.1253 \quad (4.1)$$

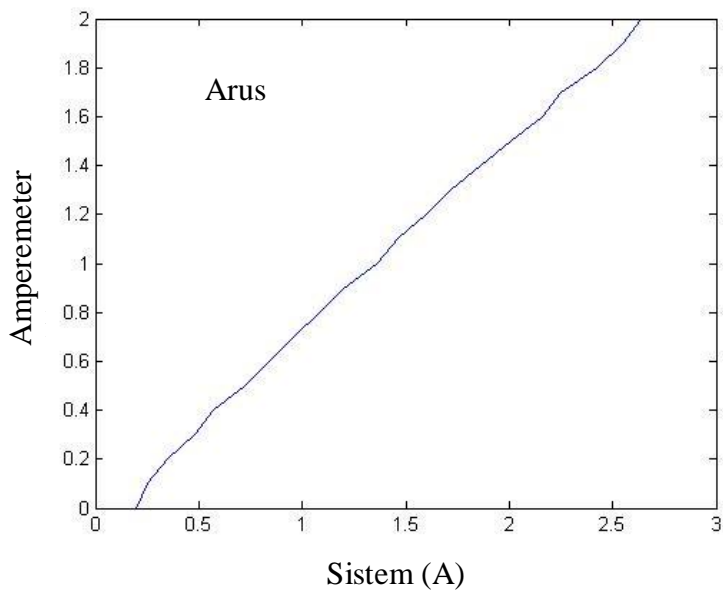
$I_r$  = Arus sebenarnya

$I_s$  = arus yang diukur oleh ACS712

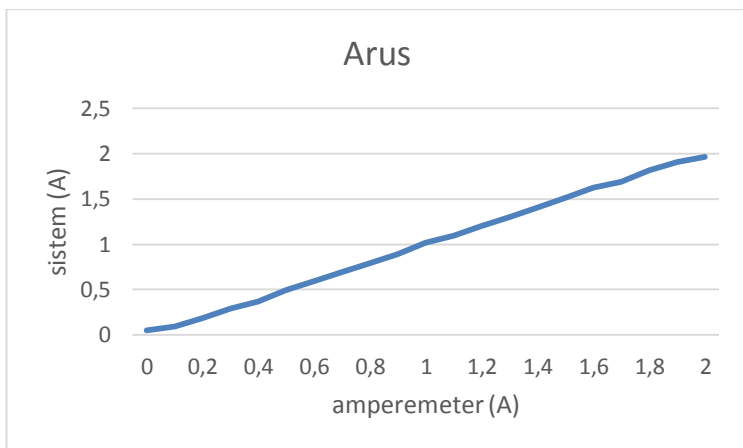
Setelah didapatkan persamaan polinomial maka nilai arus yang terbaca oleh sistem menjadi mendekati nilai yang sebenarnya seperti grafik pada Gambar 4.7.

**Tabel 4.1** Pengukuran Arus

Arus yang terbaca pada amperemeter	Arus yang terbaca pada sistem
0 A	0,2 A
0,1 A	0,25 A
0,2 A	0,35 A
0,3 A	0,48 A
0,4 A	0,57 A
0,5 A	0,72 A
0,6 A	0,84 A
0,7 A	0,96 A
0,8 A	1,08 A
0,9 A	1,2 A
1 A	1,36 A
1,1 A	1,46 A
1,2 A	1,6 A
1,3 A	1,72 A
1,4 A	1,86 A
1,5 A	2,01 A
1,6 A	2,16 A
1,7 A	2,25 A
1,8 A	2,42 A
1,9 A	2,55 A
2 A	2,64 A



**Gambar 4.7** Grafik Pengukuran Arus



**Gambar 4.8** Grafik Penghitungan Arus oleh Sistem

## 4.2 Pengukuran Suhu

Pengukuran suhu pada sistem menggunakan LM35. Pengujian dilakukan dengan membandingkan suhu yang terbaca oleh sistem dengan termometer ruangan. Arduino membaca suhu dan ditampilkan melalui *serial print* sedangkan termometer ruangan menampilkan suhu melalui display digital.

Dari pengukuran awal yang ditunjukkan oleh Tabel 4.2, hasil pengukuran suhu memiliki nilai dengan ketelitian hingga dua angka desimal. Sementara resolusi dari LM35 hanya mencapai  $1^{\circ}\text{C}$ . Hal ini dikarenakan resolusi ADC dari mikrokontroler lebih kecil daripada resolusi sensor suhu. Oleh karena itu pada program mikrokontroler nilai pengukuran suhu dibulatkan sehingga tidak ada bilangan desimal.



**Gambar 4.9** Pengukuran Suhu oleh Sistem



**Gambar 4.10** Pengukuran Suhu oleh Termometer Ruangan



**Tabel 4.2** Pengukuran Suhu

Pengukuran oleh termometer	Pengukuran oleh sistem
26 <sup>0</sup> C	25.88-26.37 <sup>0</sup> C
25 <sup>0</sup> C	24.9-25.39 <sup>0</sup> C
24 <sup>0</sup> C	23.93-24.41 <sup>0</sup> C
23 <sup>0</sup> C	22.95-23.44

**Tabel 4.3** Pembulatan Suhu

Pengukuran oleh sistem	Pembulatan
25.88-26.37 <sup>0</sup> C	26 <sup>0</sup> C
24.9-25.39 <sup>0</sup> C	25 <sup>0</sup> C
23.93-24.41 <sup>0</sup> C	24 <sup>0</sup> C
22.95-23.44	23 <sup>0</sup> C

### 4.3 Pengujian Sistem

Pengujian Sistem terdiri dari dua tahap, yaitu mengukur AC saat AC dalam keadaan kotor lalu saat AC sudah dibersihkan. Setiap tahapan ini masing-masing nantinya akan menghasilkan kecepatan pendinginan AC atau waktu yang dibutuhkan untuk menurunkan setiap <sup>0</sup>C.

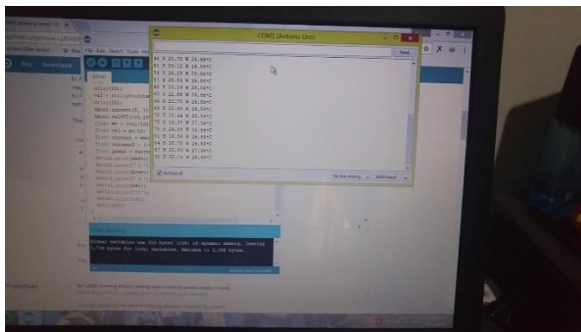
Rangkain pengukuran suhu dan arus menggunakan rangkaian yang sudah dipaparkan pada bab 3. Data akan dibaca melalui *serial print* dari IDE Arduino. Pengukuran dilakukan dengan jarak *sampling* 3 detik selama  $\pm 40$  menit



**Gambar 4.11** Tapping ACS712 pada AC



**Gambar 4.12** Pengukuran Arus dan Suhu oleh Sistem



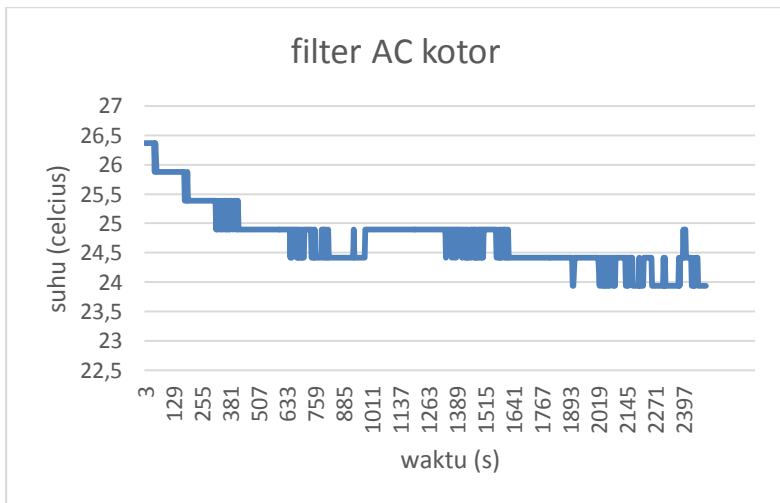
**Gambar 4.13** Data Pengukuran melalui *Serial Print*

#### 4.3.1 Pengukuran Ketika AC Kotor

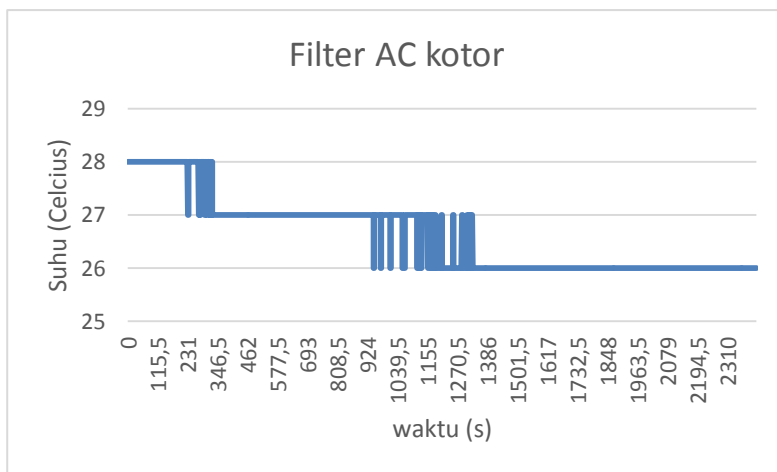
Pada tahap ini dilakukan pengukuran pada saat AC dalam keadaan kotor. AC terakhir dicuci sekitar 6 bulan sebelumnya. Pengukuran dilakukan pada dua AC yang berbeda yaitu AC Haiaer HSU-05LEA03 dan AC Sharp AH-AP5MHL. Pada AC Haier HSU-05LEA03 dilakukan pengujian dengan *fan* AC pada mode *hi* dan setpoint suhu pada *remote* adalah 20<sup>0</sup>C. Pengujian dilakukan dengan rentang waktu 40 sampai 60 menit. Grafik pengukuran pada saat AC Haier masih kotor digambarkan pada Gambar 4.14.

Pengujian pada AC Sharp AH-AP5MHL dilakukan dengan mode *fan lo, med, dan hi*. Setpoint suhu pada *remote* AC adalah  $20^{\circ}\text{C}$ . Pengujian dilakukan dengan rentang waktu 40 sampai 60 menit. Grafik pengukuran pada saat AC Sharp masih kotor dengan mode *fan lo* digambarkan pada Gambar 4.15. Grafik pengukuran pada saat AC Sharp masih kotor dengan mode *fan med* digambarkan pada Gambar 4.16. Grafik pengukuran pada saat AC Sharp masih kotor dengan mode *fan hi* digambarkan pada Gambar 4.17.

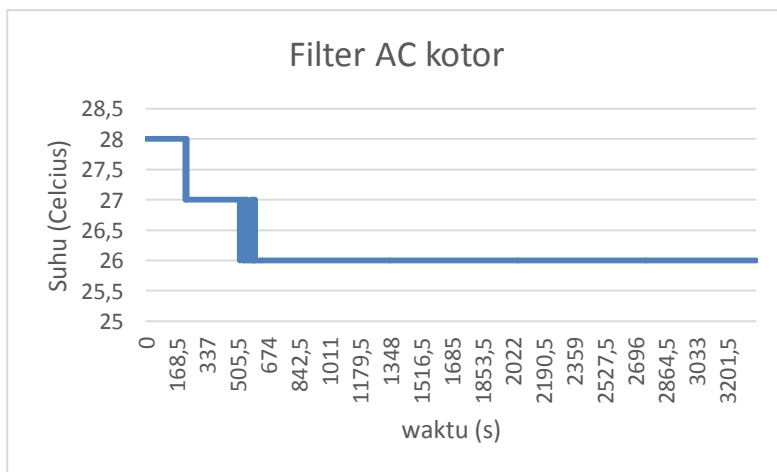
Data yang diukur kemudian dihitung sehingga didapatkan kecepatan pendinginan AC saat kotor untuk AC Haier adalah  $841.8367 \text{ s}^{\circ}\text{C}$ . Kecepatan pendinginan AC saat kotor untuk AC Sharp dengan mode *fan lo* adalah  $1462,5 \text{ s}^{\circ}\text{C}$ . Kecepatan pendinginan AC saat kotor untuk AC Sharp dengan mode *fan med* adalah  $546 \text{ s}^{\circ}\text{C}$ . Kecepatan pendinginan AC saat kotor untuk AC Sharp dengan mode *fan hi* adalah  $246 \text{ s}^{\circ}\text{C}$ .



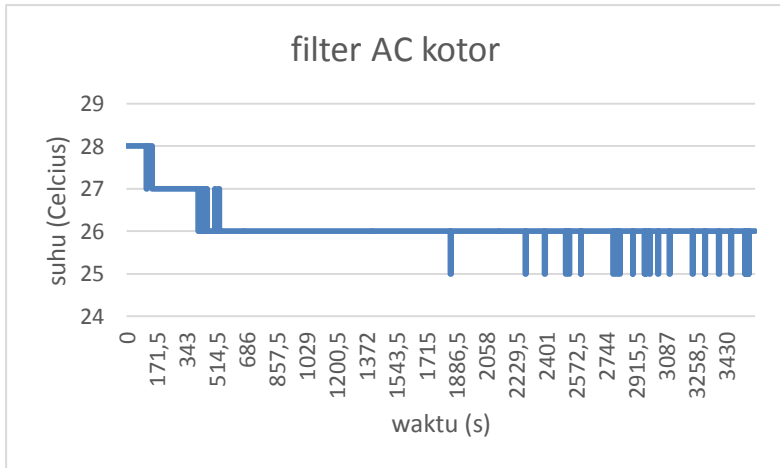
**Gambar 4.14** Pengukuran Sistem saat AC Kotor



**Gambar 4.15** Pengukuran Sistem saat AC Kotor(2)



**Gambar 4.16** Pengukuran Sistem saat AC Kotor(3)



**Gambar 4.17** Pengukuran Sistem saat AC Kotor(4)



**Gambar 4.18** Filter AC Haier kotor



**Gambar 4.19** Filter AC Sharp kotor

#### **4.3.2 Pengukuran Ketika AC Bersih**

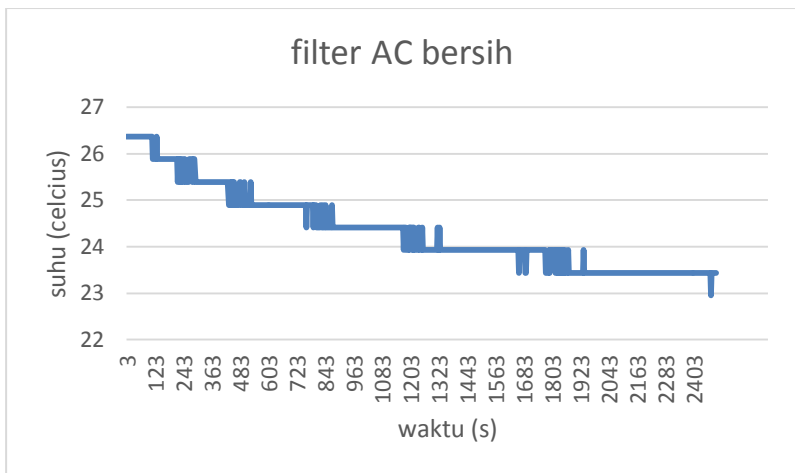
Pada tahap ini dilakukan pengukuran pada saat AC dalam keadaan bersih. Oleh karena itu sebelum dilakukan pengukuran AC dibersihkan oleh tukang servis AC. Kemudian pengujian akan diulang dengan parameter yang sama seperti saat pengujian AC dengan filter kotor. Pada AC Haier HSU-05LEA03 dilakukan pengujian dengan *fan* AC pada mode *hi* dan setpoint suhu pada *remote* adalah 20°C. Pengujian dilakukan dengan rentang waktu 40 sampai 60 menit. Grafik pengukuran pada saat AC Haier bersih digambarkan pada Gambar 4.20.

Pengujian pada AC Sharp AH-AP5MHL dilakukan dengan mode *fan lo, med*, dan *hi*. Setpoint suhu pada *remote* AC adalah 20°C. Pengujian dilakukan dengan rentang waktu 40 sampai 60 menit. Grafik pengukuran pada saat AC Sharp bersih dengan mode *fan lo* digambarkan pada Gambar 4.21. Grafik pengukuran pada saat AC Sharp bersih dengan mode *fan med* digambarkan pada Gambar 4.22. Grafik pengukuran pada saat AC Sharp bersih dengan mode *fan hi* digambarkan pada Gambar 4.23.

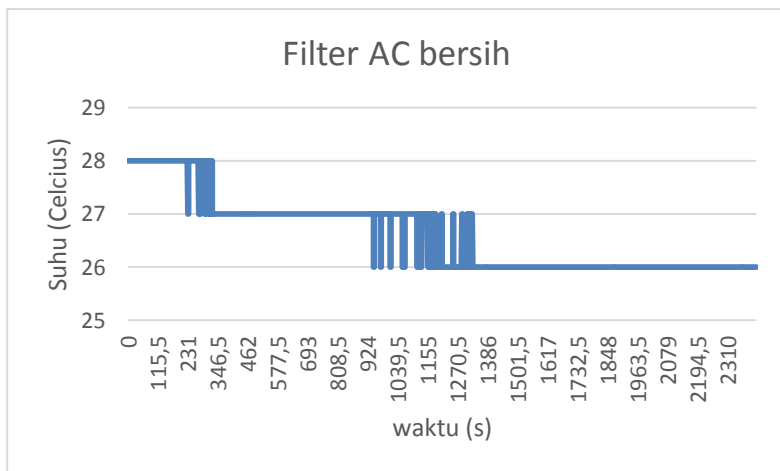
Data yang diukur kemudian dihitung sehingga didapatkan kecepatan pendinginan AC Haier saat bersih 526.2295 s<sup>0</sup>/C. Kecepatan pendinginan AC bersih untuk AC Sharp dengan mode *fan lo* adalah 621 s<sup>0</sup>/C. Kecepatan pendinginan AC bersih untuk AC Sharp dengan mode

*fan med* adalah  $358\text{ s}^{\circ}\text{C}$ . Kecepatan pendinginan AC bersih untuk AC Sharp dengan mode *fan hi* adalah  $205.5\text{ s}^{\circ}\text{C}$ .

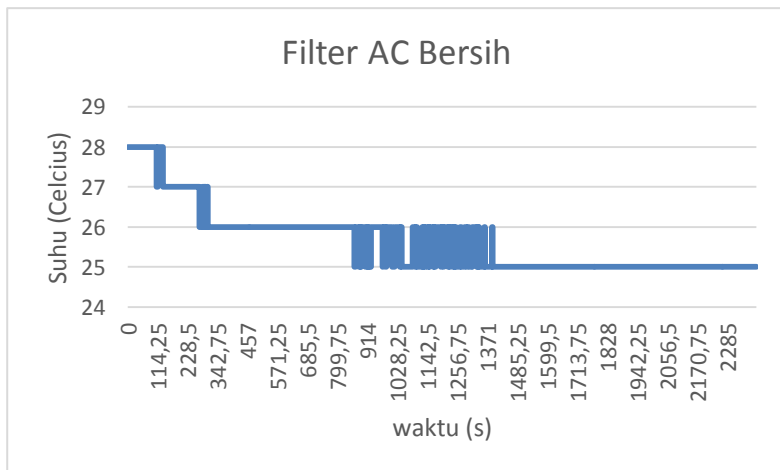
Dari pengukuran sistem saat AC kotor dan saat AC bersih, didapatkan perbedaan kecepatan pendinginan pada AC Haier sebesar  $315.6072\text{ s}^{\circ}\text{C}$ . Sedangkan perbedaan kecepatan pendinginan pada AC Sharp adalah  $841.5\text{ s}^{\circ}\text{C}$  untuk mode *fan lo*,  $188\text{ s}^{\circ}\text{C}$  untuk mode *fan med*, dan  $40.5\text{ s}^{\circ}\text{C}$  untuk mode *fan hi*. Hasil pengukuran AC kotor dan bersih menunjukkan alat mampu membedakan kondisi filter AC bersih dan kotor dengan beban dan mode fan yang berbeda.



**Gambar 4.20** Pengukuran Sistem saat AC Bersih

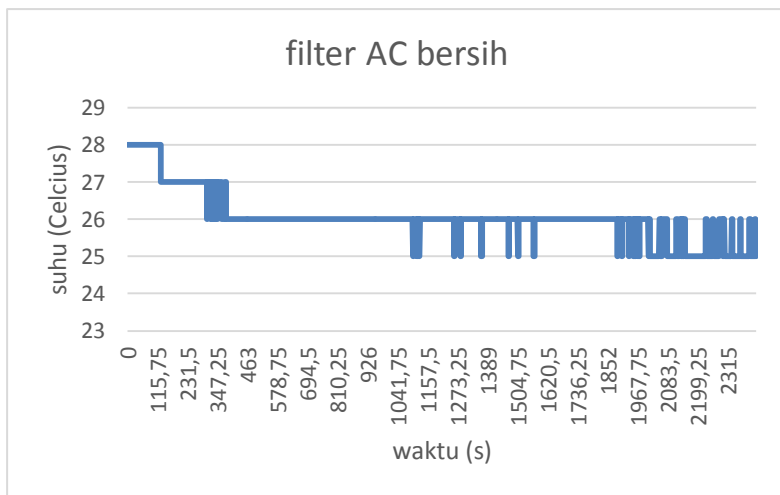


**Gambar 4.21** Pengukuran Sistem saat AC Bersih(2)



**Gambar 4.22** Pengukuran Sistem saat AC Bersih(3)





**Gambar 4.23** Pengukuran Sistem saat AC Bersih(4)



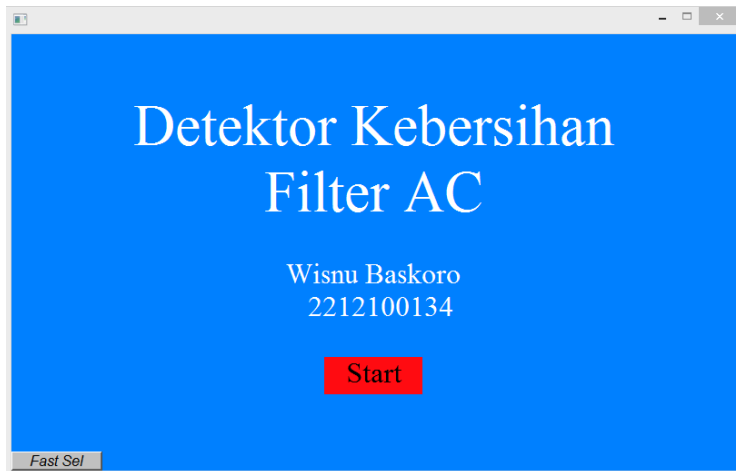
**Gambar 4.24** Proses Pencucian AC



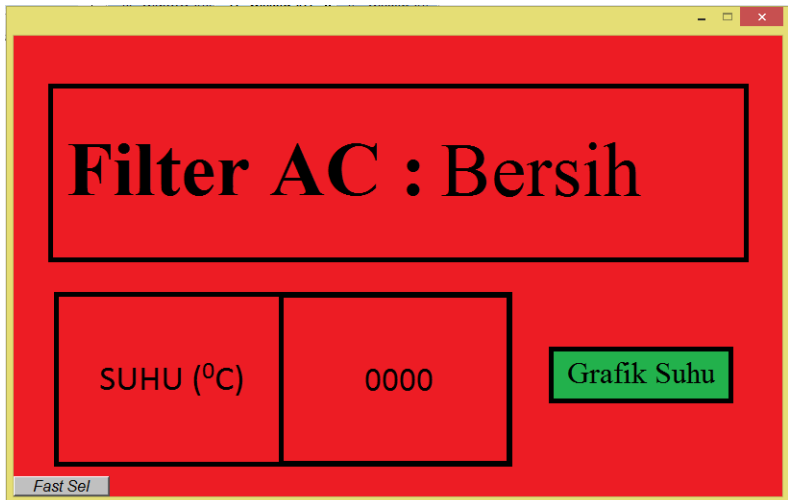
**Gambar 4.25** Filter AC bersih

#### **4.4 Pengujian HMI**

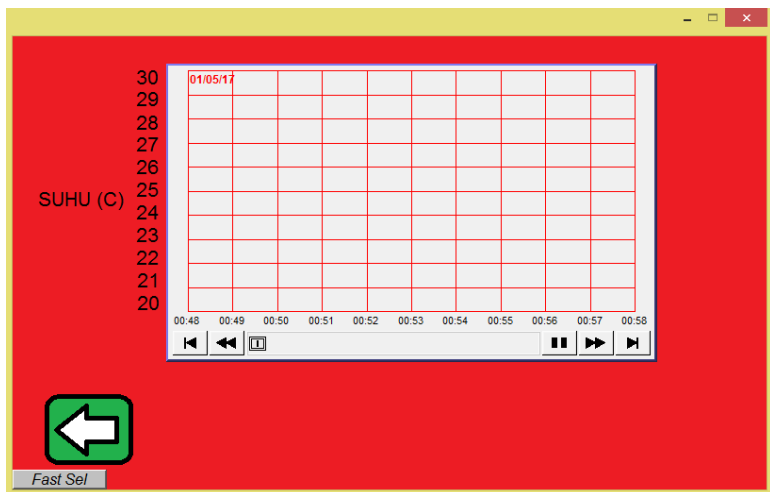
Pengujian HMI dilakukan dengan menggunakan fitur *online simulation* dari EasyBuilder8000. Komputer dihubungkan dengan sistem menggunakan kabel ethernet. Kemudian komputer akan mensimulasikan interface dari HMI. Simulasi ini mensimulasikan *interface* MT8070iH. *Screenshoot* dari simulasi HMI digambarkan pada Gambar 4.26, Gambar 4.27, dan Gambar 4.28.



**Gambar 4.26** *Online Simulation*



**Gambar 4.27** *Online Simulation (2)*



**Gambar 4.28** *Online Simulation (3)*

#### 4.5 Pengujian Sistem Menggunakan AC *Portable*

Untuk memenuhi kebutuhan Demo digunakan sebuah AC *portable* yang diletakkan dalam sebuah ruangan buatan. Tujuan dari digunakannya AC *portable* dalam ruangan buatan ini adalah untuk mempersingkat waktu demo karena ruangan buatan dibuat relatif kecil sehingga suhu akan lebih cepat turun mengingat terbatasnya waktu demo.

Pengujian dilakukan dengan memanipulasi filter AC. Filter AC akan disematkan kapas untuk menghambat sirkulasi udara layaknya debu menghambat sirkulasi udara.

Hasil Pengujian menunjukkan perbedaan waktu pendinginan antara AC dengan filter bersih dan AC dengan filter yang telah diberikan kapas. Data yang dihasilkan dari pengujian digambarkan pada Gambar 4.31 dan Gambar 4.32.

Dari pengambilan data ketika filter AC bersih menunjukkan AC memiliki kecepatan pendinginan  $37,34\text{ S}^{\circ}\text{C}$ . Dari pengambilan data ketika filter AC yang diberikan kapas menunjukkan AC memiliki kecepatan pendinginan  $70,66\text{ S}^{\circ}\text{C}$ .

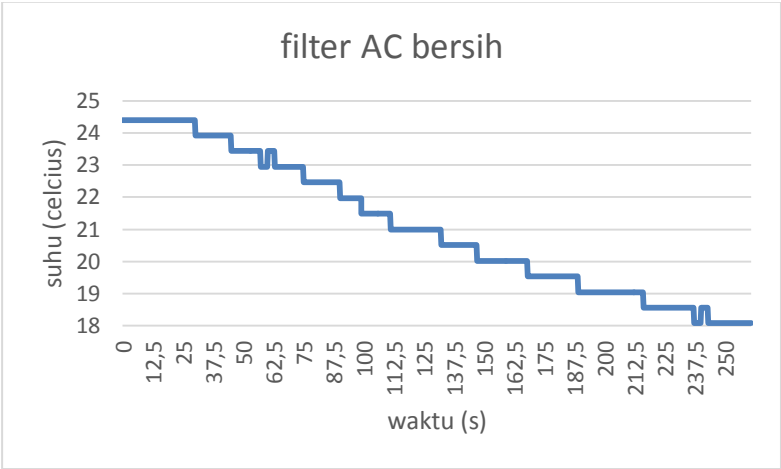
Dari kedua data pengujian di atas terdapat perbedaan kecepatan pendinginan ruangan antara AC dengan filter bersih dan AC dengan filter yang diberikan kapas  $32,32\text{ S}^{\circ}\text{C}$ .



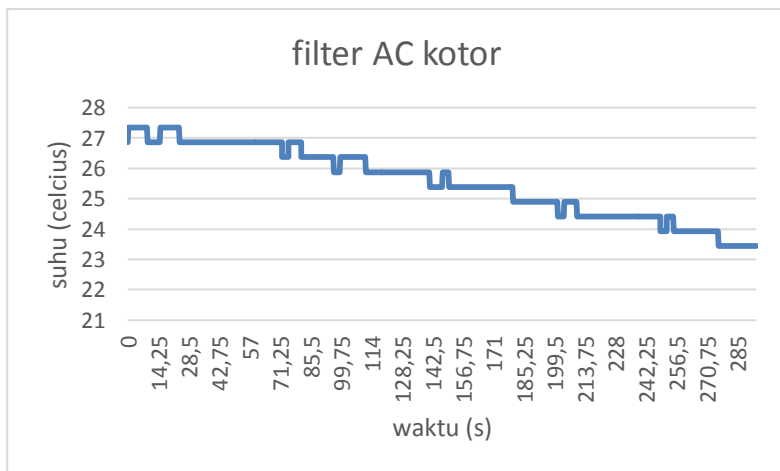
**Gambar 4.29** Filter AC Bersih



**Gambar 4.30** Filter AC Diberi Kapas



**Gambar 4.31** Grafik Pengukuran Filter AC Bersih



**Gambar 4.32** Grafik Pengukuran Filter AC Diberikan Kapas

## BAB 5

### PENUTUP

#### 5.1 Kesimpulan

Setelah melewati serangkaian tahapan tugas akhir, rancang sistem monitoring daya untuk deteksi kebersihan filter AC dapat mengukur suhu ruangan dan daya yang dikonsumsi oleh AC untuk mendeteksi kebersihan dari filter AC. Sistem akan mendeteksi apakah AC dalam kondisi menyala melalui pengukuran dayanya. Kemudian ketika AC dalam kondisi menyala, maka sistem mulai menghitung waktu yang dibutuhkan oleh AC untuk menurunkan suhu ruangan. Data yang diukur kemudian dihitung sehingga didapatkan kecepatan pendinginan AC untuk AC Haier HSU-05LEA03 saat kotor adalah  $841.8367\text{ s}^{\circ}\text{C}$ ,  $526.2295\text{ s}^{\circ}\text{C}$  saat bersih, dengan selisih  $315.6072\text{ s}^{\circ}\text{C}$ . Kecepatan pendinginan AC Sharp AH-AP5MHL dengan mode *fan lo* adalah  $1462,5\text{ s}^{\circ}\text{C}$  saat kotor dan  $621\text{ s}^{\circ}\text{C}$  saat bersih dengan selisih  $841.5\text{ s}^{\circ}\text{C}$ . Kecepatan pendinginan AC Sharp AH-AP5MHL dengan mode *fan med* adalah  $546\text{ s}^{\circ}\text{C}$  saat kotor dan  $358\text{ s}^{\circ}\text{C}$  saat bersih dengan selisih  $188\text{ s}^{\circ}\text{C}$ . Kecepatan pendinginan AC Sharp AH-AP5MHL dengan mode *fan hi* adalah  $246\text{ s}^{\circ}\text{C}$  saat kotor dan  $205.5\text{ s}^{\circ}\text{C}$  saat bersih dengan selisih  $40.5\text{ s}^{\circ}\text{C}$ .

Dari data yang didapatkan menunjukkan metode sistem mampu membedakan filter AC kotor dan bersih dengan beban dan mode *fan* AC yang berbeda. Jika waktu yang diperlukan terlalu besar, maka mikrokontroler akan memberikan indikator berupa LED dan mengirimkan data ke HMI bahwa filter dalam keadaan kotor, kemudian HMI akan memperingatkan bahwa filter AC sudah harus dibersihkan dan unit AC juga harus dibersihkan. Penggunaan alat ini dapat menghemat konsumsi listrik jika pemilik AC melakukan tindakan preventif sesuai dengan peringatan dari sistem. Jika sebuah bangunan apartemen menggunakan alat ini pada setiap kamarnya, maka akan semakin besar jumlah energi yang dihemat mengingat terbatasnya energi yang ada saat ini. Selain itu jika alat ini diterapkan pada bangunan apartemen, maka akan semakin mudah bagi pengelola bangunan untuk menjadwalkan perawatan AC untuk menjaga efisiensi konsumsi energi listrik.

## 5.2 Saran

Saran yang dapat diberikan untuk pengembangan lebih lanjut dari tugas akhir ini adalah agar sistem mampu mendeteksi kebersihan filter pada AC tipe *inverter*. Selain itu diharapkan rancangan sistem ini dapat memicu produsen-produsen AC untuk menyertakan indikator kebersihan AC mengingat berbedanya setiap kondisi tempat AC itu dipasang.



## DAFTAR PUSTAKA

- [1] Bedoyo, M. Sidik, et al. "Outlook Energi Indonesia 2015". BPPT 2015
- [2] U.S Department of Energy, "Energy-Efficient Air Conditioning", NREL, 1999
- [3] Montgomery, Ross D., "Study Verifies Coil Cleaning Saves Energy", ASHARE Journal, vol. 48, pp 34-36, November, 2006
- [4] Harianto, Feri. "Konservasi Energi Selubung Bangunan pada Gedung Graha Galaxy Surabaya". ITATS, Surabaya, 2013.
- [5] ----,----. FS-245-C. FILTERSCAN® WiFi Conduit-powered Monitor (Limit of 25), <http://cleanalert.myshopify.com/collections/catalogpro/products/fs-245-cm25-filterscan-wifi-conduit-powered-monitor>, diakses 16 Januari 2017, 06.21 WIB
- [6] Yuliastri, "Aplikasi Sensor UGN3505 sebagai Pendeteksi Medan Magnet", Elektron, vol.1, Juni, 2009
- [7] Syam, Rafiuddin, "Dasar Dasar Teknik Sensor", Fakultas Teknik Universitas Hasanuddin, Makassar, bab 2, 2013
- [8] ----,----. Thermally Sensitive Resistors, <https://www.peaksensors.co.uk/resources/thermistor-information/>, diakses 16 Januari 2017, 20.46 WIB
- [9] ----,----. Mikrokontroler AT89S51, <https://rezutopia.wordpress.com/2009/03/27/mikrokontroler-at89s51/>, diakses 16 Januari 2017, 20.49 WIB
- [10] Assa'idah dan Adnan, Yulinar, " Investigasi Terhadap Kemampuan 2 Tipe ADC", Jurnal Penelitian Sains, vol.12, no.2(B), Mei, 2009
- [11] ----,----. PIC Analog to Digital Converter Tutorial, <http://www.microcontrollerboard.com/analog-to-digital-converter.html>, diakses 16 Januari 2017, 20.51 WIB
- [12] Kharisma, Agung Chandra. "Mengeal Jaringan Metropolitan yang Didasari oleh Teknologi Ethernet" Universitas Sriwijaya, Palembang, 2015.
- [13] Fadilla, Nurin. "*Building Automation System* Berbasis Mikrokontroler Untuk Monitoring dan Kontrol Energi" ITS, Surabaya, 2015.

- [14] Hidayat, Rohmat,. dan Sumardi, “Perancangan *Human Machine Interface*(HMI) pada *Hitcut Machine* dengan PLC Omron Sysmac CP1L” UNDIP, Semarang, 2013
- [15] Daryanto, Fauzi. “Monitoring Lampu Koridor Gedung A D4 PENS-ITS dengan Menggunakan Wireless Sensor Network” PENS, Surabaya, 2011
- [16] ----,----.”Datasheet ACS712”, Allegro, 2013
- [17] ----,----. “Datasheet LM35”, Texas Instrurment, 2016
- [18] ----,----. Arduino Uno,  
<https://www.arduino.cc/en/main/arduinoBoardUno>, diakses 26 Mei 2016, 17.57 WIB
- [19] Simanjuntak, Maratur Gabe. “Perancangan Prototipe Smart Building Berbasis Arduino Uno”. Universitas Sumatera Utara, Medan, 2012.
- [20] ----,----. Haier HSU-05LEA03,  
[http://www.haier.com/id/products/air\\_conditioner/split\\_type/hsu-05lea03.shtml](http://www.haier.com/id/products/air_conditioner/split_type/hsu-05lea03.shtml), diakses 10 Januari 2017, 17.22 WIB
- [21] ----,----. AC Sharp 1/2 PK AH-AP5MHL,  
[http://lammoro.com/index.php?route=product/product&product\\_id=108](http://lammoro.com/index.php?route=product/product&product_id=108), diakses 17 Januari 2016, 07.15 WIB
- [22] ----,----. Midea MPG-09CR,  
[http://www.midea.com/id/consumer/air\\_conditioner/klassic/201504/t20150401\\_177392.shtml](http://www.midea.com/id/consumer/air_conditioner/klassic/201504/t20150401_177392.shtml), diakses 10 Januari 2017, 17.38 WIB

## LAMPIRAN

Program IDE Arduino

```
#include "EmonLib.h" // Include Emon Library
EnergyMonitor emon1; // Create an instance
#include <Mudbus.h>
#include <SPI.h>
#include <Ethernet.h>
Mudbus Mb;
int val;
int val2=0;
int valcount=0;
int valreal=0;
int tempPin = 0;
float tempstart=0;
float tempbuff=0;
int tempcount = 0;
float waktu = 0;
int waktubuff = 0;
float cooling=0;
float current2=0;
float currentcount=0;
float currentreal=0;
int kondisi=0;
int bersih=35;
int kotor=50;
float loss=0;

void setup()
{
  uint8_t mac[]   = { 0x90, 0xA2, 0xDA, 0x00, 0x51, 0x06 };
  uint8_t ip[]    = { 192, 168, 1, 8 };
  uint8_t gateway[] = { 192, 168, 1, 1 };
  uint8_t subnet[] = { 255, 255, 255, 0 };
  Ethernet.begin(mac, ip, gateway, subnet);
  //Avoid pins 4,10,11,12,13 when using ethernet shield
  delay(5000); //Time to open the terminal
  Serial.begin(9600);
  emon1.current(5, 10); // Current: input pin, calibration.
```

```

}

void loop()
{
  val = analogRead(tempPin); // read temperature adc value
  delay(10);
  val = analogRead(tempPin); // read temperature adc value
  delay(10);
  emon1.calcVI(10,200); // Calculate all. No.of wavelengths, time-out
  val2=val2+val;
  valcount++;
  if(valcount==12){
    valreal=val2/12;
    val2=0;
    valcount=0;
  }
  float mv = (valreal/1024.0)*5000; //calculate temperature
  float cel = mv/10;
  float current = emon1.Irms; //extract Irms into Variable
  current2=current2+current;
  currentcount++;
  if(currentcount==12){
    currentreal=current2/12;
    current2=0;
    currentcount=0;
  }
  float          currentcalibrated          =          (-
0.0374*currentreal*currentreal)+(0.8912*currentreal)-0.1253;
//eliminate error from sensor
  float power = currentcalibrated*220; //calculate power asuming voltage
is 220vac
  if (power > 150){
    waktu=waktu+0.25; //time counter
    if(waktu<4){
      tempstart = cel; //starting temperature record
      tempbuff = cel;
    }
    if(cel>tempstart){
      tempstart=cel;

```

```

    }
    if(cel<tempbuff){ //calculate cooling speed
        tempbuff = cel;
        waktubuff = waktu;
        cooling = waktubuff/(tempstart - tempbuff);
    }
}
else{
    waktu=0;
    if(cooling>=kotor){
        kondisi=1;
    }
    if(cooling<kotor){
        kondisi=0;
    }
}
if(cooling>bersih){
    loss=(cooling-bersih)*900;
}
Mb.Run();
Mb.R[0]=cel;
Mb.R[3]=kondisi;
delay(10);
}

```

*# Halaman ini sengaja dikosongkan #*

## BIODATA PENULIS



Wisnu Baskoro dilahirkan di Cirebon, pada tanggal 11 Januari 1994 dari pasangan Bapak Doddy Priambodo dan Ibu Rina Erliati. Penulis adalah anak kedua dari tiga bersaudara. Anak pertama adalah Imaniar Swariandina dan anak ketiga adalah Danang Wicaksono.

Perjalanan akademik penulis dimulaik sejak umur enam tahun masuk SDN Polisi 1 Bogor. Setelah lulus SD penulis melanjutkan sekolahnya di SMPN 1 Bogor. Setelah lulus SMP penulis melanjutkan sekolahnya di SMAN 1 Bogor. Setelah lulus SMA penulis melanjutkan studinya di Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya. Judul tugas akhir penulis adalah "*Sistem Monitoring Waktu Pendinginan Ruangan untuk Deteksi Kebersihan Filter AC*".

Email: wisnubaskorowisnu@gmail.com

*# Halaman ini sengaja dikosongkan #*